

# Zeitoptimale Trajektorienplanung für automatisiertes Fahren bis in den fahrdynamischen Grenzbereich

Ingmar Gundlach





# Zeitoptimale Trajektorienplanung für automatisiertes Fahren bis in den fahrdynamischen Grenzbereich

Vom Fachbereich  
Elektrotechnik und Informationstechnik  
der Technischen Universität Darmstadt  
zur Erlangung des Grades eines Doktor-Ingenieurs (Dr.-Ing.)  
genehmigte Dissertation

von

**Dipl.-Ing. Ingmar Helge Gundlach**

geboren am 11. September 1986

Referent: Prof. Dr.-Ing. Ulrich Konigorski  
Korreferent: Prof. Dr.-Ing. Boris Lohmann

Tag der Einreichung: 13.03.2020  
Tag der Prüfung: 17.07.2020



# Zeitoptimale Trajektorienplanung für automatisiertes Fahren bis in den fahrdynamischen Grenzbereich

Gemehmigte Dissertation von Dipl.-Ing. Ingmar Helge Gundlach

D 17 · Darmstadt 2020

Dieses Dokument wird bereitgestellt von TUprints, E-Publishing-Service der TU Darmstadt  
<https://tuprints.ulb.tu-darmstadt.de>  
[tuprints@ulb.tu-darmstadt.de](mailto:tuprints@ulb.tu-darmstadt.de)

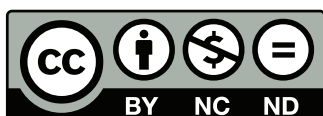
URN: urn:nbn:de:tuda-tuprints-191201

URL: <https://tuprints.ulb.tu-darmstadt.de/19120>

Diese Publikation ist auch in gebundener und elektronischer Form beim Shaker Verlag, Düren verfügbar:

ISBN: 978-3-8440-7573-1

DOI: 10.2370/9783844075731



Die Veröffentlichung steht unter folgender Creative-Commons-Lizenz:

Namensnennung – Nicht-kommerziell – Keine Bearbeitung 4.0 International

<https://creativecommons.org/licenses/by-nc-nd/4.0/deed.de>

# Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Fachgebiet Regelungstechnik und Mechatronik (RTM) der TU Darmstadt im Rahmen eines Kooperationsprojekts mit der Konzernforschung der Volkswagen AG.

Mein besonderer Dank gilt meinem Doktorvater Herrn Prof. Dr.-Ing. Ulrich Konigorski für die hervorragende Betreuung. Als Fachgebietsleiter ließ er den Mitarbeitern Freiraum, brachte ihnen viel Vertrauen entgegen und nahm sich stets Zeit, um über Fragen oder Probleme zu sprechen, was zu einer überaus angenehmen Arbeitsatmosphäre führte. Außerdem danke ich Herrn Prof. Dr.-Ing. Boris Lohmann für die Übernahme des Korreferats und das Interesse an meiner Arbeit.

Ganz herzlich bedanke ich mich beim gesamten (aktuellen und ehemaligen) RacePilot-Team von VW für die unübertreffliche Zusammenarbeit, insbesondere bei Dr. Paul Hochrein, Dr. Jens Hoedt, Jonas Kaste, Dr. Björn Mennenga und Dr. Kristof van Ende. Neben den zahlreichen sehr konstruktiven und zielgerichteten fachlichen Diskussionen werde ich mich natürlich immer voller Freude an die wunderbaren Erprobungsfahrten mit einem großartigen Team erinnern. Vielen Kollegen aus dem Team bleibe ich freundschaftlich verbunden. Außerdem gilt mein Dank Thomas Behrens von der KST innovations GmbH für seine große Hilfsbereitschaft bei der Software-Integration und Inbetriebnahme sowie Dr. Dirk Schütte von der IAV GmbH für seine Unterstützung u. a. während der Erprobungen.

Am Fachgebiet RTM, wo ich meinen Arbeitsplatz hatte, habe ich eine angenehm familiäre Atmosphäre unter den Kollegen genossen. Das schließt auch die Mitarbeiterinnen des Sekretariats Corina Fischer und Ilse Brauer sowie Brigitte Hoppe, deren Verlust uns alle hart getroffen hat, mit ein. Durch ihre vorbildliche Organisation und Unterstützung trugen sie wesentlich dazu bei. Selbiges gilt auch für den Systemadministrator Alfred Gross, der in jeder Angelegenheit äußerst hilfsbereit und unkompliziert unterstützte. Ein besonderer Dank gilt Dr. Eric Lenz, der sich wie kaum ein anderer umgehend in eine Thematik eindenken und fundierte Ratschläge geben kann. Ihm wie auch Dr. Jens Hoedt danke ich zudem für das Korrekturlesen dieser Arbeit und Dr. Jakob Bechtloff für die interessanten Gespräche über Fahrdynamik.

Nicht zuletzt gilt mein inniger Dank meinen Eltern, die mir zunächst mein Studium der Elektro- und Informationstechnik ermöglichten und mich in vielerlei Hinsicht all die Jahre liebevoll unterstützten.

Ingolstadt, August 2020

*Ingmar Gundlach*



# Inhaltsverzeichnis

<b>Formelzeichen und Abkürzungen</b>	<b>VIII</b>
<b>Kurzfassung</b>	<b>XIV</b>
<b>Abstract</b>	<b>XVI</b>
<b>1 Einführung</b>	<b>1</b>
1.1 Zielsetzung und Anwendungsgebiete . . . . .	2
1.2 Projektbeschreibung . . . . .	4
1.2.1 Versuchsträger . . . . .	4
1.2.2 Implementierung . . . . .	5
1.2.3 Systemarchitektur . . . . .	5
1.3 Nomenklatur und Notation . . . . .	7
1.4 Gliederung der Arbeit . . . . .	9
<b>2 Konzeption der Trajektorienplanung</b>	<b>10</b>
2.1 Definition des Lösungsraums . . . . .	10
2.2 Wechselwirkung von Längs- und Querplanung . . . . .	14
2.3 Zeit- oder Weg-Parametrierung . . . . .	18
2.4 Literaturüberblick zur Rundenzeitoptimierung . . . . .	19
<b>3 Modellbildung und Zeitoptimalität</b>	<b>25</b>
3.1 Minimierung der Rundenzeit mittels Weg-Zeit-Transformation . . . . .	25
3.1.1 Kinematik des Fahrzeugs . . . . .	27
3.1.2 Die Referenzlinie als Bezugssystem der Fahrlinie . . . . .	28
3.2 Modellierung der Fahrzeugdynamik . . . . .	29
3.2.1 Reifenmodell . . . . .	29
3.2.2 Nichtlineares Einspurmodell . . . . .	34
3.2.3 Normalkraft und Achslastverteilung . . . . .	40
3.2.4 Dynamische Bremskraftverteilung . . . . .	42
3.2.5 Simulation zum Einfluss der Achslast auf die Kraftschlussgrenze . . . . .	44
3.3 Zusammenfassung . . . . .	47
<b>4 Methoden der nichtlinearen Optimierung</b>	<b>48</b>
4.1 Diskretisierung eines dynamischen Optimierungsproblems . . . . .	49
4.2 Statische Optimierung ohne Nebenbedingungen . . . . .	50
4.2.1 Bestimmung der Suchrichtung . . . . .	51
4.2.2 Liniensuche . . . . .	53

4.3	Statische Optimierung mit Nebenbedingungen . . . . .	54
4.3.1	Gleichungsnebenbedingungen . . . . .	54
4.3.2	Ungleichungsnebenbedingungen . . . . .	56
4.4	Nichtlineares Interior-Point-Verfahren Ipopt . . . . .	59
4.4.1	Inertia Correction und Feasibility Restoration Phase . . . . .	61
4.4.2	Filter Line-search und Second-Order-Correction . . . . .	61
4.4.3	Adaptive Barriere-Strategie . . . . .	62
<b>5</b>	<b>Optimierungsproblem</b>	<b>64</b>
5.1	Diskretisierung und Aktordynamik . . . . .	65
5.1.1	Modellierung der Aktordynamik . . . . .	66
5.1.2	Formulierung der Aktordynamik als quadratische Straffunktion . . . . .	69
5.1.3	Berechnung diskreter Zeitableitungen höherer Ordnung . . . . .	70
5.2	Gütemaß der Trajektorienplanung . . . . .	74
5.2.1	Sekundäres Gütemaß: Straffunktion für Gleichungsnebenbedingungen . . . . .	74
5.2.2	Sekundäres Gütemaß: Straffunktion für Ungleichungsnebenbedingungen . . . . .	75
5.3	Nebenbedingungen der Trajektorienplanung . . . . .	80
5.3.1	Box-Restriktionen . . . . .	81
5.3.2	Nichtlineare Nebenbedingungen . . . . .	81
5.4	Implementierung . . . . .	85
5.5	Ergebnisse der Rundkurs-Planung . . . . .	88
5.5.1	Zusammenfassung . . . . .	88
5.5.2	Simulationsergebnisse für das Autodrom Most . . . . .	89
5.6	Erweiterung zur „Komfort-Planung“ . . . . .	94
5.6.1	Komfort-Potential . . . . .	95
5.6.2	Ergebnisse . . . . .	97
<b>6</b>	<b>Toolchain der Trajektorienplanung</b>	<b>99</b>
6.1	Erstellung der Referenzlinie . . . . .	100
6.1.1	Preprocessing der Fahrbahnränder . . . . .	100
6.1.2	Sicherheitsabstand . . . . .	101
6.1.3	Berechnung der Mittellinie . . . . .	105
6.1.4	Krümmungsabhängige Abtastung der Referenzlinie . . . . .	108
6.2	Ein Kappa-Filter zur Glättung parametrierter Kurven . . . . .	109
6.2.1	Filterung durch Optimierung . . . . .	110
6.2.2	Formulierung eines quadratischen Optimierungsproblems . . . . .	113
6.2.3	Ergebnisse des Kappa-Filters zum Glätten . . . . .	116
6.2.4	Ergebnisse des Kappa-Filters zur Rennlinien-Approximation . . . . .	116
6.3	Matching-Algorithmus zur Referenzierung eines Punktes auf einem Spline . . . . .	123
6.3.1	Spline-Zell-Suche (cell search) . . . . .	124
6.3.2	Analytische Bestimmung des Match-Punkts . . . . .	125



<b>7 Zyklische Echtzeit-Planung mit Objekten</b>	<b>127</b>
7.1 Echtzeit- und Rechenzeitanforderungen . . . . .	127
7.2 Optimierungsproblem bei gleitendem Horizont . . . . .	128
7.3 Statische Objekte . . . . .	131
7.4 Ergebnisse der Echtzeit-Planung . . . . .	132
7.4.1 Simulationsergebnisse der zeitoptimalen Planung . . . . .	133
7.4.2 Simulationsergebnisse zu Ausweichmanövern in der Komfort-Planung . .	139
7.4.3 Messergebnisse einer Realfahrt auf dem Autódromo do Algarve . . . . .	139
<b>8 Zusammenfassung und Ausblick</b>	<b>143</b>
<b>A Gleitkommazahlen nach IEEE 754</b>	<b>147</b>
<b>B Autobahnquerschnitte und Kurvenradien</b>	<b>148</b>
<b>C Differenzenquotient</b>	<b>149</b>
C.1 Zentraler Differenzenquotient . . . . .	150
C.2 Vorwärts-Differenzenquotienten . . . . .	150
<b>D Ableitungen im Optimierungsproblem</b>	<b>152</b>
D.1 Gradient und HESSE-Matrix der Gütefunktion . . . . .	152
D.2 JACOBI-Matrix der Nebenbedingungen . . . . .	153
D.2.1 Systemgleichungen . . . . .	153
D.2.2 Kammscher Kreis . . . . .	154
D.2.3 Maximale Längskraft durch die Motorleistung . . . . .	154
D.2.4 Reifenkräfte und Schräglaufwinkel . . . . .	154
D.3 HESSE-Matrizen der Nebenbedingungen . . . . .	155
D.3.1 Systemgleichungen . . . . .	155
D.3.2 Kammscher Kreis . . . . .	157
D.3.3 Maximale Längskraft durch die Motorleistung . . . . .	157
D.3.4 Reifenkräfte und Schräglaufwinkel . . . . .	157
<b>E Box-Restriktion der Längskraft</b>	<b>159</b>
<b>Literaturverzeichnis</b>	<b>161</b>
<b>Eigene Veröffentlichungen</b>	<b>169</b>
<b>Betreute studentische Arbeiten</b>	<b>170</b>

# Formelzeichen und Abkürzungen

## Lateinische Formelzeichen

Zeichen	Beschreibung	Einheit
$a$	Beschleunigung	$\text{m s}^{-2}$
$A$	Querspanntfläche	$\text{m}^2$
$B$	PACEJKA-Reifenmodell-Parameter	1
$C$	PACEJKA-Reifenmodell-Parameter	1
$\mathbf{c}(\mathbf{x}_{\text{opt}})$	Funktion der Nebenbedingungen im Optimierungsproblem	
$C_{\text{acc}}, C_{\text{dec}}$	Funktion zur Definition des Komfort-Gütemaßes	
$c_w$	Luftwiderstandsbeiwert	1
$c_\theta c_\phi$	$c_\theta c_\phi := \cos(\theta) \cos(\phi)$	1
$d$	Querversatz zur Referenzlinie	m
$D$	PACEJKA-Reifenmodell-Parameter	N
$\mathbf{D}_n$	Matrix für den $n$ -ten Differenzenquotienten nach dem Weg	$\text{m}^{-n}$
$ds$	Streckeninkrement projiziert auf die Referenzlinie	m
$ds_t$	Streckeninkrement parallel zur Referenzlinie	m
$ds_v$	Streckeninkrement in Richtung des Geschwindigkeitsvektors	m
$E$	PACEJKA-Reifenmodell-Parameter	1
$f(\mathbf{x}_{\text{opt}})$	Gütefunktion eines statischen Optimierungsproblems	
$F_B$	Reifen-/Achslängskraft durch die Betriebsbremse	N
$F_{Lx}$	Luftwiderstand in Fahrzeuginnenrichtung	N
$F_M$	Reifen-/Achslängskraft durch den Antrieb	N
$F_{\text{max},r}$	Maximal übertragbare Horizontalkraft am Reifen / an der Achse	N
$F_{x_r}$	Umfangskraft am Reifen / an der Achse	N
$F_{y_r}$	Seitenkraft am Reifen / an der Achse	N
$F_w$	Widerstandskraft	N
$F_{z_r}$	Aufstandskraft / Achslast	N
$g$	Erdbeschleunigung $g = 9,81 \text{ m s}^{-2}$	$\text{m s}^{-2}$
$G$	Gesamte Gewichtskraft in $z$ -Richtung	N
$\mathbf{g}(\mathbf{x}_{\text{opt}})$	Funktion für Gleichungsnebenbedingungen	
$h$	Schrittweite entlang der Referenzlinie	m
$\mathbf{h}(\mathbf{x}_{\text{opt}})$	Funktion für Ungleichungsnebenbedingungen	
$h_{\text{SP}}$	Schwerpunkthöhe	m
$\mathbf{i}, \mathbf{I}$	Einsvektor, Einheitsmatrix	1
$J$	Gütefunktional	
$\mathbf{J}_f(\mathbf{x})$	JACOBI-Matrix einer Funktion $f(\mathbf{x})$ , siehe Notation auf S. 8	
$j_x$	Längsruck	$\text{m s}^{-3}$

Zeichen	Beschreibung	Einheit
$J_z$	Trägheitsmoment des Fahrzeugs um die $z$ -Achse	$\text{kg m}^2$
$k_{\text{BV}}, k_{\text{MV}}$	Bremskraft- bzw. Motorkraftverteilung	1
$k_{\text{Fz}}$	Degressivitätsfaktor für den Einfluss der Achslast	1
$k_{\text{Lx}}$	Proportionalitätsfaktor des Luftwiderstands $k_{\text{Lx}} = \frac{1}{2} c_w A \rho_L$	$\text{kg m}^{-1}$
$l$	Radstand	m
$\mathcal{L}$	LAGRANGE-Funktion	
$l_h, l_v$	Abstand Gesamtschwerpunkt – Hinter- bzw. Vorderachse	m
$m, m_h, m_v$	Masse des Fahrzeugs insgesamt / Hinterachse / Vorderachse	kg
$\vec{n}$	Normalenvektor	
$N$	Anzahl der Streckensegmente bei diskretisiertem Streckenverlauf	1
$N_{\text{opt}}$	Anzahl der Optimierungsvariablen	1
$\mathbf{p}$	<i>Nur Kapitel 4:</i> Suchrichtung	
$\vec{p}, P$	Ortsvektor bzw. Punkt im kartesischen Koordinatensystem	
$p_{\text{acc}}, p_{\text{dec}}$	Parameter im Komfort-Gütemaß für Längsbeschleunigung	%
$P_{\text{M}}$	Antriebsleistung	W
$q$	Parameter im Komfort-Gütemaß für Querschleunigung	%
$r$	Radius	m
$s$	$\left\{ \begin{array}{l} \text{Bogenlänge (ohne Index: von der Referenzlinie)} \\ \text{Nur bei Übertragungsfunktionen: LAPLACE-Variable} \end{array} \right.$	$\text{m}$ $\text{rad s}^{-1}$
$\mathbf{s}$	<i>Nur Kapitel 4:</i> Schlupfvariable	
$S$	Wegstrecke	m
$s_r$	Reifenschlupf	1
$t$	Zeit	s
$T$	Zeitdauer	s
$\mathbf{u}$	Stellgrößenvektor	
$v$	Geschwindigkeit	$\text{m s}^{-1}$
$w_{\text{acc}}, w_{\text{dec}}$	Gewichtung im Komfort-Gütemaß	s
$x$	Fzg-feste Koordinate vom SP in Fahrzeuginnenrichtung	m
$\mathbf{x}$	Zustandsvektor	
$x_0$	Globale Koordinate	m
$\mathbf{x}_{\text{opt}}$	Vektor der Optimierungsvariablen	
$x_r$	Radfeste Koordinate in Radlängsrichtung	m
$y$	Fzg-feste Koordinate vom SP horizontal, senkrecht zur $x$ -Achse	m
$y_0$	Globale Koordinate horizontal, senkrecht zur $x_0$ -Achse	m
$y_r$	Radfeste Koordinate horizontal, senkrecht zur $x_r$ -Achse	m
$z$	Fzg-feste Koordinate vom SP vertikal nach oben	m
$z_B$	Abbremsung $z_B = -a_x/g$	1

## Griechische Formelzeichen

Zeichen	Beschreibung	Einheit
$\alpha$	$\left\{ \begin{array}{l} \text{Schräglaufwinkel} \\ \text{Nur Kapitel 4: Schrittweite bei Liniensuche} \end{array} \right.$	$\text{rad}$ 1
$\beta$	Schwimmwinkel	rad
$\gamma$	Winkel zwischen $x_0$ -Koordinate und Referenzlinie	rad
$\delta$	Lenkwinkel	rad
$\varepsilon$	Maschinengenauigkeit	
$\zeta$	Parameter bei der Diskretisierung der Referenz (Abschnitt 6.1.4)	1
$\theta$	Fahrbahnneigungswinkel in Längsrichtung	rad
$\vartheta$	Funktion für den MAYER-Term	
$\kappa$	Krümmung (ohne Index: von der Referenzlinie)	$\text{m}^{-1}$
$\lambda$	Gewichtungsfaktor im Gütemaß	
$\lambda$	Nur Kapitel 4: LAGRANGE-Multiplikator	
$\mu$	$\left\{ \begin{array}{l} \text{Kraftschlussbeiwert} \\ \text{Nur Kapitel 4: Barriereparameter} \end{array} \right.$	1 1
$\rho$	Radius der Bahnkurve	m
$\rho_L$	Luftdichte	$\text{kg m}^{-3}$
$\sigma$	Umrechnungsfaktor $\sigma = dt/ds$	$\text{s m}^{-1}$
$\sigma_\mu$	Nur Kapitel 4: Centering-Parameter der Barrierefunktion	1
$\tau$	Zeitkonstante	s
$\phi$	Fahrbahnneigungswinkel in Querrichtung	rad
$\varphi$	Gütefunktion in einem dynamischen Optimierungsproblem	
$\Phi$	Gütefunktion in einem statischen Optimierungsproblem	
$\chi$	Kurswinkel	rad
$\psi$	Gierwinkel	rad
$\psi_t$	Auf die Referenz bezogener Gierwinkel $\psi_t = \psi - \gamma$	rad

## Häufig verwendete Indices

Index	Beschreibung
++	Erweitert einen Vektor $\mathbf{x} \in \mathbb{R}^N$ um den Abtastwert $x_{N+1}$ , sodass $\mathbf{x}_{++} \in \mathbb{R}^{N+1}$
Ab	(Strecken-)Abschnitt (im Gegensatz zu Rk für Rundkurs)
acc	acceleration (positive Längsbeschleunigung)
B	Bremse
c	Zentripetal(-beschleunigung) / Komfort-Gütemaß
dec	deceleration (Verzögerung)
end	Endwert
h	Hinterachse
krit	kritischer Wert
L	Luft bzgl. Aerodynamik

Index	Beschreibung
lap	Rundenzeit, Fahrzeit (lap time)
lb	Untergrenze (lower bound)
li	links
M	Motor, Antrieb
max	Maximaler Wert
n	normiert (bei Vektoren)
opt	Optimierungsvariable
r	Rad / Reifen
re	rechts
ref	Referenzlinie
Rk	Rundkurs (im Gegensatz zu Ab für Abschnitt)
rl	Fahrlinie (raceline)
sys	Funktion im Fahrzeugmodell
t	Tangential zur Referenzlinie
ub	Obergrenze (upper bound)
v	Vorderachse
w	Widerstandskraft
x, y, z	Koordinatenrichtung in Fahrzeugkoordinaten
$x_r, y_r, z_r$	Koordinatenrichtung in Radkoordinaten

## Mathematische Symbole und Operatoren

Symbol	Bedeutung
$\ \mathbf{x}\ _n$	$\ell_n$ -Norm von $\mathbf{x}$
$\mathbf{x} \odot \mathbf{y}$	Elementweise Multiplikation (HADAMARD-Produkt) von $\mathbf{x}$ und $\mathbf{y}$
$\mathbf{x}^{oi}$	Elementweise $i$ -te Potenz von $\mathbf{x}$ , siehe Gl. (5.15) auf S. 71
$x^*$	Optimaler Wert der Größe $x$
$\bar{x}$	Arithmetisches Mittel einer Folge $(x_i)$
$\hat{x}$	Maximum von $x$
$\dot{x}$	Ableitung von $x$ nach der Zeit
$x'$	Ableitung von $x$ nach dem Weg
$x^{(n)}$	$n$ -te Ableitung von $x$ nach dem Weg
$\Delta$	Differenz zweier Werte: <ul style="list-style-type: none"> <li>• Mit Index als Vorwärtsdifferenz: <math>\Delta x_k := x_{k+1} - x_k</math></li> <li>• Bei Messdaten oder im Regelkreis: <math>\Delta x := x_{\text{soll}} - x_{\text{ist}}</math></li> </ul>
$\frac{\Delta^n x}{\Delta s^n}$	Differenzenquotient $n$ -ter Ordnung von $x$ nach $s$
$\nabla f$	Gradient von $f$ , siehe Notation auf S. 8
$\nabla^\top f$	Liegender Gradient von $f$ , siehe Notation auf S. 8
$\nabla^2 f$	HESSE-Matrix von $f$ , siehe Notation auf S. 8
$f \circ g$	Komposition/Verkettung zweier Funktionen: $(f \circ g)(x) = f(g(x))$

**Abkürzungen**

ACC	Adaptive Cruise Control
AD	Automatisches Differenzieren
ADTF	Automotive Data and Time-Triggered Framework
ALU	Arithmetic Logic Unit
AVX	Advanced Vector Extensions
BKV	Bremskraftverteilung
BLAS	Basic Linear Algebra Subprograms
CPU	Central Processing Unit
DGL	Differentialgleichung
DGPS	Differential Global Positioning System
DoF	Degree of Freedom (Freiheitsgrade)
DP	Diskretisierungspunkt
E/A	Eingang/Ausgang
ESP	Elektronisches Stabilitätsprogramm
FAS	Fahrerassistenzsystem
Fzg	Fahrzeug
GNB	Gleichungsnebenbedingung
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPU	Graphics Processing Unit
INS	Inertial Navigation System
IP	Interior-Point(-Methode)
Ipoint	Interior Point Optimizer [86]
KKT	KARUSH-KUHN-TUCKER(-Bedingungen)
LAPACK	Linear Algebra Package
LGS	Lineares Gleichungssystem
LICQ	Linear Independence Constraint Qualification
LKA	Lane Keeping Assistant
MBK	Motor-/Brems-Kontravalenz (Komplementärbedingung)
MIMD	Multiple Instruction, Multiple Data
MIMO	Multiple Input, Multiple Output
MKL	Math Kernel Library von Intel® [44]
MPC	Model Predictive Control
NB	Nebenbedingung
o. B. d. A.	Ohne Beschränkung der Allgemeinheit
QP	Quadratisches Programm (Optimierungsproblem)
QSS	Quasi-steady state (quasi-statisch)
SIMD	Single Instruction, Multiple Data
SISO	Single Input, Single Output
SKF	Super-Kappa-Filter
SP	Schwerpunkt

SQP	Sequentiell quadratische Programmierung
SSE	Streaming SIMD Extensions
TTC	Time To Collision
u. B. v.	Unter Beschränkung von
UNB	Ungleichungsnebenbedingung

# Kurzfassung

Um ein Gesamtsystem zum automatisierten Fahren bis in den fahrdynamischen Grenzbereich zu entwickeln und zu testen, eignet sich eine zeitoptimale Trajektorienplanung am besten, weil die Zeitoptimalität erfordert, dass sich die Trajektorie permanent an einer physikalischen Grenze befinden muss. Situationen am fahrdynamischen Limit können nicht nur bei plötzlichen Ausweichmanövern auftreten, sondern auch durch Witterungseinflüsse wird das Limit mitunter schon bei moderaten Fahrmanövern erreicht.

Die vorliegende Dissertation beschreibt die Entwicklung und Umsetzung einer Trajektorienplanung, die unter Berücksichtigung fahrdynamischer Eigenschaften von Serienfahrzeugen die Fahrzeit durch numerische nichtlineare Optimierung minimiert. Drei wesentliche Aspekte liegen dem Konzept zugrunde: Erstens wird der Lösungsraum der Planung nicht a priori auf vorgegebene Trajektorien oder Bewegungsmuster eingeschränkt. Zweitens erfolgen Quer- und Längsplanung simultan, also in einer gemeinsamen Optimierung. Und drittens wird als unabhängiger Parameter der Differentialgleichungen nicht die Zeit, sondern die Bogenlänge einer Referenzlinie, die entlang der Fahrbahn verläuft, verwendet. Dafür werden alle zeitabhängigen Gleichungen anhand kinematischer Beziehungen transformiert. Diese nichtlineare Transformation ermöglicht es, ein Gütefunktional aufzustellen, das die tatsächliche Fahrzeit entlang der Wegstrecke beschreibt. So muss kein Optimierungsproblem mit freier Endzeit gelöst werden. Zudem hat sie den Vorteil, dass statische Elemente, wie der Fahrbahnverlauf oder die Position der Leitplanken, im mathematischen Problem statisch bleiben.

Als Fahrzeugmodell kommt ein nichtlineares Einspurmodell mit einer PACEJKA-Reifenkennlinie zum Einsatz, welches um eine Achslast- und Bremskraftverteilung erweitert ist. Die Aktordynamik (Motor, Bremse, Lenkung) wird über einen speziellen Ansatz abgebildet, der im Optimierungsproblem die Nebenbedingungen in Abhängigkeit der Aktorausgänge angibt, was die Dimension des Optimierungsproblems reduziert. Zudem werden diese Nebenbedingung in Straffunktionen mit einer geschwindigkeitsabhängigen „semi-quadratischen“ Form umgewandelt. Der Ansatz erhöht nicht nur die Konvergenz, sondern führt auch zu sehr glatten Trajektorien, deren Krümmung bis mindestens zur zweiten Ableitung stetig ist, was man im Fahrverhalten deutlich spürt.

Da die Trajektorienplanung sowohl offline zur Berechnung eines gesamten Rundkurses als auch zur echtzeitfähigen zyklischen Planung unter Berücksichtigung statischer Hindernisse eingesetzt wird, liegt der Fokus auf einer effizienten Umsetzung, was sich sowohl in der Modellbildung und Problemformulierung als auch der Implementierung widerspiegelt. So wird bspw. ein 4,5 km langer Rundkurs mit IPOPT auf einer Mobile-CPU in unter 2 s berechnet. Die zyklische Planung benötigt bei einem krümmungsabhängigen Planungshorizont von mindestens 126 m bzw. 6,6 s (testweise) durchschnittlich 40 ms. Somit ist diese Trajektorienplanung schneller als vergleichbare Ansätze aus der Literatur.



Eine Eigenschaft der dargelegten zeitoptimalen Planung ist die harmonische Verbindung von Quer- und Längsdynamik. Daher ist sie zum einen perfekt geeignet, um optimale Brems-/Ausweich-Kombinationen zu berechnen. Zum anderen bietet sie auch eine gute Basis für eine komfort-orientierte Trajektorienplanung, welche als Erweiterung der bestehenden Planung entwickelt wurde.

Im Rahmen der Toolchain der Trajektorienplanung wurde u. a. ein Algorithmus entwickelt, der mittels quadratischer Programmierung verrauschte Kurvenverläufe innerhalb vorgegebener Grenzen glättet. Er wird primär zum Glätten der Referenzlinie eingesetzt, kann aber auch sehr gut zur approximativen Berechnung von Ideallinien verwendet werden.

Die Trajektorienplanung wurde mit der Volkswagen AG Konzernforschung, Wolfsburg auf diversen Rennstrecken und Handlings-Kursen erfolgreich erprobt, wie auch die abgebildeten Messdaten zeigen.

# Abstract

In order to develop and test a complete system for automated driving up to the driving dynamic limit, time-optimal trajectory planning is best suited. This is because time optimality requires the trajectory to be permanently at a physical limit. Situations at the limit of driving dynamics do not only occur during sudden evasive maneuvers: the limit is sometimes also reached during moderate driving maneuvers e.g. due to weather influences.

The present thesis describes the development and implementation of a trajectory planning algorithm, which minimizes the driving time by numerical nonlinear optimization, taking into account the dynamic properties of series production vehicles. The concept is based on three essential aspects: Firstly, the solution space for the planning is not limited a priori to given trajectories or movement patterns. Secondly, lateral and longitudinal planning are performed simultaneously, i.e. in a combined optimization. And thirdly, the independent parameter of the differential equations is not the time, but the arc length of a reference line along the track. For this purpose, all time-dependent equations are transformed using kinematic relations. This nonlinear transformation enables to set up an objective functional that describes the actual driving time along the path. This optimization problem can be solved without free end time. It also has the advantage that static elements, such as the course of the road or the position of the guard rails, remain static in the mathematical problem.

The vehicle model used is a non-linear single-track model with a PACEJKA magic tyre formula. It is extended by an axial load and brake force distribution. The actuator dynamics (engine, brake, steering) are mapped using a special approach which specifies the constraints in the optimization problem as a function of the actuator outputs. Thus, the dimension of the optimization problem is reduced. In addition, these constraints are converted into penalty functions that have a velocity-dependent “semi-quadratic” form. This approach not only increases the convergence, but also leads to very smooth trajectories, whose curvature is continuous up to at least the second derivative. The smoothness is significantly noticeable in the driving behavior.

Since trajectory planning is used both offline for the calculation of an entire circuit and for real-time moving horizon planning taking static obstacles into account, the focus is on efficient implementation. This is reflected in the modelling and problem formulation as well as in the implementation. As a result, a 4.5 km long circuit can be calculated with IPOPT on a mobile CPU in less than 2 s. The moving horizon planning requires an average of 40 ms with a curvature-dependent planning horizon of at least 126 m or 6.6 s, respectively. Thus, this trajectory planning algorithm is faster than comparable approaches from the literature.

One characteristic of the proposed time-optimal planning is the harmonious combination of lateral and longitudinal dynamics. Therefore it is perfectly suited for calculating optimal brake/evasion

combinations. On the other hand, it also offers a good basis for a comfort-oriented trajectory planning, which was developed as an extension of the existing planning.

Within the toolchain of the trajectory planning an algorithm was developed, which smoothes noisy curves within given limits by quadratic programming. It is primarily used for smoothing the reference line, but can also be used very well for the approximate calculation of racing lines.

The trajectory planning was successfully tested with Volkswagen AG Group Research, Wolfsburg on various race tracks and handling parkours, as the measurement data presented in this thesis also demonstrates.



# 1 Einführung

Aktuell ist die Entwicklung der Automobilindustrie stark von Systemen zum teil- und vollautomatisierten Fahren geprägt. Systeme, die einzelne Fahrfunktionen übernehmen, wie z. B. das Adaptive-Cruise-Control (ACC) zur Längsführung oder die Lane Keeping Assistance (LKA) zur Querrführung, sind bereits auf dem Markt etabliert. Da diese Systeme den Fahrer lediglich unterstützen und ihm nicht die Verantwortung für das sichere Führen seines Fahrzeugs abnehmen, werden sie als Fahrerassistenzsysteme (FAS) bezeichnet. Dem Fahrer obliegt die Aufgabe, die Systeme permanent auf ihr korrektes Verhalten zu überwachen und gegebenenfalls unverzüglich einzugreifen. Der Funktionsumfang des ACC beispielsweise unterliegt zudem engen Grenzen: Die ISO 22179 sieht bei Geschwindigkeiten über 20 m/s (72 km/h) lediglich Beschleunigungen unter  $2 \text{ m/s}^2$  und Verzögerungen unter  $3,5 \text{ m/s}^2$  vor [91, Kap. 46.3.2].

Der Grad der Automatisierung wächst zurzeit kontinuierlich, indem zum einen Funktionalitäten bestehender Systeme kombiniert werden und zum anderen der Funktionsumfang oder die Menge der unterstützten Szenarien zunimmt. So kombiniert ein Stauassistent Funktionalitäten aus ACC und LKA, indem er einem vorausfahrenden Fahrzeug in Längsrichtung und dem Fahrstreifen in Querrichtung folgt. Bei so einem *teilautomatisierten System* nach SAE-Level 2 trägt der Fahrer noch immer die Überwachungsaufgabe [47]. Die Funktion ist zudem noch auf niedrige Geschwindigkeiten begrenzt (z. B. bei Volkswagen aktuell bis 60 km/h). Mit solchen Systemgrenzen können nicht alle auftretenden Szenarien im realen Straßenverkehr abgedeckt werden und es ist daher notwendig, dass der Fahrer als Rückfallebene zur Verfügung steht. Selbst bei *bedingter Automatisierung* nach Level 3 muss der Fahrer das System zwar nicht ständig überwachen, jedoch als Rückfallebene zur Verfügung stehen, auch wenn ihm hier eine gewisse Zeit zur Übernahme gewährt werden muss [47]. Erst ab Level 4 oder 5 muss eine Sicherheitsarchitektur, wie z. B. in [69] beschrieben, vorhanden sein, mit der Systemfehler soweit ausgeschlossen werden können, dass man auf die Fahrer-Rückfallebene verzichtet.

Es ist eine „Ironie der Automatisierung“, dass der Mensch in gewöhnlichen Szenarien durch eine Maschine ersetzt wird, aber in Situationen, in denen die Maschine versagt, also in besonders kritischen Situationen, doch auf sich allein gestellt ist [4]. Neben einem Komfortgewinn durch Automatisierung soll i. A. die Sicherheit erhöht werden, indem der fehlerbehaftete, unzuverlässige Mensch durch eine Maschine ersetzt wird, die viel komplexere Berechnung innerhalb kürzester Zeit reproduzierbar durchführen kann. Jedoch soll derselbe unzuverlässige Mensch die Maschine auf Fehler überwachen, obwohl er den komplexen Berechnungen doch so schnell gar nicht folgen kann [4]. In [62] wird eine solche Übernahmeproblematik anhand einer Probandenstudie untersucht: Sie zeigt Probleme bei der Übernahme auf, wenn der Fahrer aufgrund unzureichenden Systemverständnisses den Systemzustand fehlinterpretiert oder zu hohes Vertrauen in das System hat. Dabei wurden zusätzliche Schwierigkeiten durch eine „Deaktivierung“ des Fahrers nach lan-

ger Unterforderung [73, Kap. 5.5] oder durch fahrdynamisch anspruchsvolle Situationen noch gar nicht berücksichtigt.

Zum einen kann diese Problematik entschärft werden, indem der Fahrer bestmöglich über den Systemzustand informiert und ihm ein ausreichendes Systemverständnis vermittelt wird [62, 73, 4]. Zum anderen müssen aber die Systemgrenzen auch schon bei teilautomatisiertem Fahren (Level 2) und erst recht bei bedingter Automatisierung (Level 3) deutlich über das hinausgehen, was in gewöhnlichen Szenarien verlangt wird. Denn nur so kann zumindest teilweise sichergestellt sein, dass der Fahrer nicht plötzlich in einer kritischen Situation auf sich allein gestellt ist. Es sind dabei viele Systemgrenzen zu beachten, wie Grenzen von Umfeldsensorik, Planung, Regelung, Aktorik, Fahrdynamik sowie von der Systemarchitektur (z. B. Redundanz). Oft ist der Aufwand, eine signifikante Verbesserung in einem dieser Bereiche zu erzielen, jedoch immens. Ob und wann durch den steigenden Grad der Automatisierung im Straßenverkehr das vielzitierte Ziel, die Unfallzahlen zu senken, erreicht wird, lässt sich erst einige Jahre nach deren Einführung rückwirkend durch eine Statistik auf Grundlage einer ausreichenden Anzahl automatisiert fahrender Fahrzeuge feststellen. Nichtsdestoweniger lohnt es sich, an der Umsetzung dieses Ziels zu arbeiten, und an Verbesserungen für das automatisierte Fahren zu forschen.

Die vorliegende Arbeit beschäftigt sich mit der Trajektorienplanung im fahrdynamischen Grenzbereich. Während der fahrdynamische Grenzbereich auf Stabilisierungsebene durch aktive Sicherheitssysteme wie ESP schon lange abgedeckt wird, findet er auf der übergeordneten Bahnplanungsebene eher wenig Beachtung. Zwar gibt es Systeme, wie Notbremsassistenten, Ausweich-Lenk-Assistenten oder beides in Kombination wie [9, 90], welche sich aber häufig auf einzelne Szenarien oder kurzfristige Manöver fokussieren. Und weil man mit automatisiertem Fahren primär komfortable Fahrmanöver assoziiert, die jegliche kritische Situationen durch ihre Vorausschau vermeiden, werden Trajektorienplaner oft so ausgelegt, dass sie hohe Beschleunigungen a priori ausschließen. Doch ist für teilautomatisierte und besonders hochautomatisierte Fahrzeuge sicheres Planen und Fahren bis in den fahrdynamischen Grenzbereich sehr wichtig. Diese Arbeit kann dazu beitragen, die Systemgrenzen zu erweitern, indem mehr Szenarien abgedeckt werden können und auch solche, die durch die sensorische Vorausschau und eine Car2X-Kommunikation zu spät erfasst werden, um darauf komfortabel reagieren zu können. Und letztendlich entschärft es die Übernahmeproblematik, da das System in der Lage ist, länger selbst zu agieren.

## 1.1 Zielsetzung und Anwendungsgebiete

Situationen am fahrdynamischen Limit können nicht nur durch plötzliche Ausweichmanöver zur Kollisionsvermeidung auftreten; bei regennasser Fahrbahn oder erst recht bei Schnee- und Eisglätte wird das Limit schon bei moderaten Fahrmanövern erreicht. Um ein Gesamtsystem von der Planung über die Regelung bis hin zur Umsetzung zu entwickeln und zu testen, eignet sich eine zeitoptimale Trajektorie am besten, weil sie sich permanent an einer physikalischen Grenze befindet. Darum ist die Rennstrecke auch das härteste Testfeld für die Serie [76]. In der vorliegenden Arbeit ist das primäre Ziel der zu planenden Trajektorie daher stets die Minimierung der Fahrzeit,

obwohl die eigentliche Intention weniger in der Minimierung der Rundenzeit auf der Rennstrecke als vielmehr in der Generierung der so entstehenden Trajektorien liegt.

Die Berechnung einer zeitoptimalen Trajektorie ohne Berücksichtigung von Hindernissen kann bei bekanntem Streckenverlauf offline erfolgen. Im Gegensatz dazu muss beispielsweise aufgrund unvorhersehbarer Objekte eine Neuberechnung online innerhalb einiger Millisekunden erfolgen. Der hier vorgestellte Ansatz wird für Offline- und zyklische Online-Planungen gleichermaßen eingesetzt und kann daher sowohl kurze Streckenabschnitte als auch mehrere Kilometer lange Rundkurse berechnen. In den ersten Kapiteln dieser Arbeit wird der allgemeine Ansatz daher unabhängig von dem Einsatz als Offline- oder Online-Planer beschrieben. Anschließend wird auf die zusätzlichen Herausforderungen einer zyklischen Online-Planung eingegangen. Durch die Nutzung zur Online-Planung ist eine recheneffiziente Umsetzung essentiell. Sie ist ein K.-o.-Kriterium für den Einsatz und daher eines der wichtigsten Ziele dieses Projekts, was sich in vielen Aspekten dieser Arbeit widerspiegelt.

Um zeitoptimale Trajektorien im fahrdynamischen Grenzbereich in der gewünschten Güte zu berechnen, müssen Nichtlinearitäten im Modell berücksichtigt werden und auch der Planungshorizont muss hinreichend lang sein. Allerdings gilt modellbasierte Trajektorienplanung mittels nichtlinearer Optimierung für gewöhnlich auch auf PCs als zu rechenaufwendig, um sie in Echtzeit mit einem längeren Planungshorizont einzusetzen. Deswegen greift man oft auf lineare Modelle oder einen recht kurzen Horizont von bspw. 2 Sekunden in [90] zurück, wenn überhaupt ein modellbasiertes Optimierungsproblem zum Einsatz kommen soll. In dieser Arbeit wird gezeigt, dass eine modellbasierte Trajektorienplanung mittels nichtlinearer Optimierung durchaus in einer Online-Planung eingesetzt werden kann, wenn der Algorithmus von der Problemformulierung und dem Entwurf an bis zur Programmierung auf niedrige Rechenzeiten getrimmt wird.

Einer der wesentlichen Aspekte bei der zyklischen Online-Planung ist, dass aufgrund der begrenzten Rechenzeit nur noch ein Streckenabschnitt anstelle der gesamten Strecke berechnet werden kann. Mit diesem begrenzten Vorausschauhorizont allein ist die Berechnung einer zeitoptimalen Ideallinie unmöglich. Ein Ziel dieser Arbeit besteht darin, eine zwar suboptimale aber möglichst gute Lösung hierfür zu finden, wozu auch Informationen des restlichen Streckenverlaufs verwendet werden. Auch ein professioneller Fahrer wird auf einer ihm unbekannten Strecke in seiner ersten Runde niemals eine Bestzeit fahren. Denn der Streckenverlauf hinter der übernächsten Kurve kann sich stark auf die Ideallinie am aktuellen Ort auswirken. Im schlimmsten Fall ist die aktuelle Geschwindigkeit zu hoch, um das Fahrzeug stabil durch alle weiteren Kurven zu bringen.

Neben der erwähnten Entwicklung von Systemen für automatisiertes Fahren bis ans fahrdynamische Limit gibt es weitere Einsatzgebiete von Rennlinienberechnungen: Vermehrt werden automatisierte Rennwagen für Autorennen entwickelt, wie bei ROBORACE (s. [10]) oder der FORMULA STUDENT DRIVERLESS (s. [30]). Außerdem kann die Trajektorienplanung in Schulungssystemen, die den Fahrer das Fahren auf Rennstrecken lehren, wie dem VOLKSWAGEN GOLF R RACE-TRAINER [70, 71] oder dem BMW TRACK TRAINER [88], eingesetzt werden. Natürlich ist hierbei nicht nur die Trajektorienplanung, sondern auch die Fahrerinteraktion wichtig. Ein weiteres häufig genutztes Anwendungsgebiet von Rennlinienberechnungen ist auch die Optimierung von



(a) VW Golf 7 GTI „Norbert“



(b) Audi RS 7 „Walter“

**Bild 1.1:** Versuchsträger zur Erprobung des Systems (Fotos Volkswagen / Martin Meiners).

Fahrzeugparametern bei der Fahrzeugkonzeptionierung zur Verbesserung der Rundenzeit (vgl. Literaturüberblick in Abschnitt 2.4). Während in der vorliegenden Arbeit seriennahe Fahrzeuge betrachtet werden, handelt es sich dort meist um Formel-Wagen oder „Supersportwagen“.

Auch wenn in dieser Arbeit fast ausschließlich das Fahren im Grenzbereich betrachtet wird, kann der vorgestellte Ansatz mit gewissen Modifikationen auch als Basis einer Trajektorienplanung im normalen Straßenverkehr dienen. Diese Zielsetzung führt zu einem modifizierten Gütemaß, welches im Kompromiss zur Rundenzeit möglichst komfortable Verläufe generiert.

## 1.2 Projektbeschreibung

Die Arbeiten zu dieser Dissertation fanden im Rahmen eines Kooperationsprojekts zwischen dem Fachgebiet Regelungstechnik und Mechatronik der Technischen Universität Darmstadt und der Volkswagen AG Konzernforschung, Wolfsburg statt.

### 1.2.1 Versuchsträger

Als Versuchsträger kamen zwei automatisierte Versuchsträger, ein VW Golf 7 GTI (Bild 1.1(a)) und ein Audi RS 7 (Bild 1.1(b)) zum Einsatz. Zur Berechnung der automatisierten Fahrfunktionen stehen Embedded-PCs mit INTEL Core i-Prozessoren und eine dSpace AutoBox zur Verfügung. Zur Lokalisierung sowie zur Messung des Bewegungs- und Dynamikprofils werden im Fahrzeug (Fzg) Messdaten aus einer Inertialsensorik (INS) mit einem globalen Navigationssystem (GNSS) fusioniert. Als GNSS wird ein differentielles globales Positionierungssystem (DGPS) verwendet, bei dem die Genauigkeit des GPS durch Korrekturdaten erhöht wird. Die Korrekturdaten stammen von einer Referenzstation in der Nähe des Fahrzeugs, deren Position bekannt ist (z. B. mittels Langzeitmessung ermittelt). Die Referenzstation berechnet aus den aktuellen GPS-Daten und ihrer eigenen Position die tatsächlichen Signallaufzeiten für die Satelliten und sendet sie an den DGPS-Empfänger.



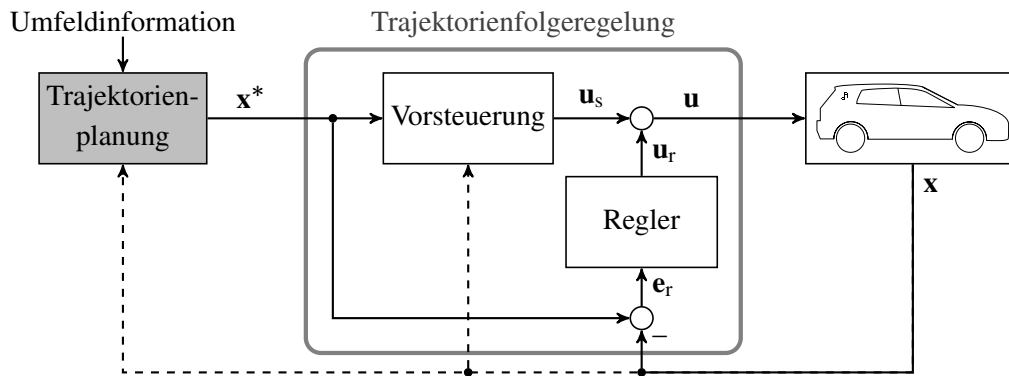


Bild 1.2: Trajektorienplanung und Regelung sind strukturell getrennt.

### 1.2.2 Implementierung

Die Implementierung der Forschungsergebnisse erfolgte sowohl in MATLAB als auch in C++. Die Trajektorienplanung wurde in ein ADTF-Framework<sup>1</sup> integriert und auf verschiedenen Test- und Rennstrecken erprobt. Die Trajektorienplanung kann sowohl offline zur Berechnung der Ideallinie für einen gesamten Rundkurs als auch zur zyklischen Online-Planung eingesetzt werden. Im Rahmen dieses Forschungsprojekts wurden statische Objekte online berücksichtigt, für dynamische Objekte wurden Ansätze entwickelt, welche aber erst in einem aktuell laufenden Nachfolgeprojekt dahingehend weiterentwickelt und verbessert werden, dass auch hierfür Online-Berechnungen und reale Testfahrten möglich sind.

### 1.2.3 Systemarchitektur

Die Architektur sieht eine strukturelle Trennung von Trajektorienplanung und Trajektorienfolgeregelung gemäß Bild 1.2 vor. Die Trajektorienplanung löst ein Optimierungsproblem mit der Fahrzeit als (primärem) Gütemaß unter Einhaltung der Differentialgleichungen eines nichtlinearen Fzg-Modells. Zur Berücksichtigung der fahrdynamischen Grenzen kommen weitere Nebenbedingungen, wie der KAMMSCHE KREIS an Vorder- und Hinterachse sowie eine Begrenzung der Antriebskraft durch die vorhandene Motorleistung hinzu. Das Optimierungsproblem liefert für einen gegebenen Streckenverlauf eine Trajektorie, also sowohl die Bahn als auch die Geschwindigkeit sowie weitere fahrdynamische Größen. Hierfür benötigt das Optimierungsproblem keine spezielle Startlösung. Die berechnete Trajektorie  $\mathbf{x}^*$  wird an eine Trajektorienfolgeregelung gegeben, die mittels *exakter Eingangs-/Ausgangs-Linearisierung* entworfen wurde. Bei der exakten E/A-Linearisierung wird die nichtlineare Regelstrecke durch eine nichtlineare Rückführung exakt in ein lineares System überführt, für welches dann mit den Methoden der linearen Regelungstechnik ein Regler entworfen werden kann [2]. Die Methode der exakten E/A-Linearisierung ist u. a.

<sup>1</sup>EB ASSIST ADTF [25] (Automotive Data and Time-Triggered Framework) ist ein Software-Tool zum Entwickeln, Validieren, Visualisieren und Testen von Fahrerassistenz- und automatischen Fahrfunktionen. Es kombiniert eine Entwicklungsumgebung mit einer interaktiven Arbeitsumgebung. Das von der Audi Electronics Venture (AEV) entwickelte Tool wird seit 2008 unter der Bezeichnung „EB Assist“ von der Elektrobit Automotive (EB) vertrieben.

in [1, Kap. 5.2] (für SISO- und MIMO-Systeme), [46, Kap. 5.2] (nur MIMO) und [2] (nur SISO) beschrieben. Die Regelung in Bild 1.2 besteht aus einer Vorsteuerung, die auf Basis eines inversen Modells den Stellgrößenverlauf  $\mathbf{u}_s$  so berechnet, dass das Fahrzeug unter idealen Bedingungen (keine Störungen, keine Modellfehler) gerade dem gewünschten Verlauf folgt. Kommt es zu einer Abweichung  $\mathbf{e}$  zwischen der Sollgröße  $\mathbf{x}^*$  und dem tatsächlichen Verhalten  $\mathbf{x}$ , überlagert der Regler dem Steuerungsanteil  $\mathbf{u}_s$  die Stellgröße  $\mathbf{u}_r$ . Die Gesamtstruktur entspricht der einer klassischen *Zwei-Freiheitsgrade-Regelung* [29, S. 9 f.]. Jedoch ist die Steuereinrichtung hier in die Trajektorienplanung und die eigentliche Vorsteuerung aufgeteilt. Um trotz etwaiger Abweichungen zur Solltrajektorie korrekte Vorsteuersignale generieren zu können, muss die Vorsteuerung zumindest Teile des Fahrzeugzustands  $\mathbf{x}$ , wie Position und Geschwindigkeit, kennen. Die vorliegende Arbeit bezieht sich ausschließlich auf die Planung; die Regelung ist nicht Bestandteil dieser Arbeit.

Die Trennung von Planung und Regelung hat einige Vorteile gegenüber einer klassischen modellprädiktiven Regelung (MPC). Bei der MPC wird zwar ebenfalls eine Trajektorie durch Lösen eines Optimierungsproblems ermittelt, indem das Systemverhalten anhand eines Modells prädiziert wird, jedoch wird dabei auch der Stellgrößenverlauf  $\mathbf{u}$  berechnet, welcher direkt zur Regelung des Prozesses verwendet wird [1, S. 439 ff.]. So erfolgen Planung und Regelung simultan. Durch die Trennung ist es möglich, die Regelung mit kürzeren Zykluszeiten auszuführen als die Planung, sodass der Regler schneller auf Störungen reagieren kann. Die Planung ist üblicherweise rechenintensiver und muss zwangsläufig langsamer laufen. Wie in Bild 1.2 zu erkennen ist, benötigt die Planung außerdem durch die Kaskadenstruktur lediglich ein Modell des *geschlossenen* unterlagerten Regelkreises, was die Modellkomplexität i. A. reduziert. Zudem können die Modelle in Planung und Regelung unterschiedlich angepasst werden, indem z. B. sehr schnelle Dynamiken, die nur für die Regelung relevant sind, in der Planung ggf. ignoriert werden.

Des Weiteren entsteht durch diese Trennung ein zusätzlicher Freiheitsgrad, wie das System mit Abweichungen  $\mathbf{e} \neq \mathbf{0}$  umgehen kann: Planung und/oder Regler können darauf reagieren. Hier ist die Aufteilung so, dass die Reduzierung der Querablage und des Orientierungsfehlers ausschließlich dem Regler obliegt. Entsteht eine zeitliche Abweichung, weil das Fahrzeug schneller oder langsamer als geplant gefahren ist, wird sie im Regler und der Planung durch einen Matching-Algorithmus abgefangen, welcher in Abschnitt 6.3 beschrieben ist. Wie die Strategie aussieht und welche Vor- und Nachteile sich daraus ergeben, wird in Abschnitt 7.2 beschrieben. Ein Nachteil der Trennung von Planung und Regelung ist der etwas höhere Entwicklungsaufwand, weil nicht nur beide Komponenten entworfen, implementiert, appliziert und getestet werden müssen, sondern auch eine Gesamtabstimmung erfolgen muss, damit ihr Zusammenspiel gut funktioniert.

Die Zwei-Freiheitsgrade-Struktur, die durch die Aufteilung von Planung mit Vorsteuerung und Regelung entsteht, entspricht auch dem *Zwei-Ebenen-Modell für das Fahrerlenkverhalten* nach Donges [22, Kap. 2.3]. Er beschreibt die Führungsebene als „antizipatorische Steuerung“ und die Stabilisierungsebene als „kompensatorische Regelung“. Dabei weisen Fahrer eine typische Antizipationszeit von ca. 1 s auf. Schnellere Ereignisse können nicht mehr auf der Führungsebene, sondern nur auf der Stabilisierungsebene behandelt werden [22, Kap. 2.4]. Auch wenn sich die Zahlenwerte nicht auf ein technisches System übertragen lassen, sieht man, dass die Unterteilung in eine schnellere Regelung und eine langsamere Planung dem menschlichen Verhalten entspricht.

## 1.3 Nomenklatur und Notation

Im Folgenden werden einige häufig verwendete Begriffe und mathematische Notationen erläutert.

### Ideallinie und krümmungsminimale Linie

Der Begriff *Ideallinie* wird, wenn nicht explizit anders formuliert, für diejenige *Fahrlinie* verwendet, auf der ein Fahrzeug die Strecke in kürzest möglicher Zeit (*zeitoptimal*) zurücklegen kann. Sie beschreibt somit eine fahrzeugspezifische Linie. Im Gegensatz dazu beschreibt [88] die klassische Ideallinie als diejenige Linie, „bei deren Befahren mit gleichbleibender Geschwindigkeit die geringsten Fliehkräfte entstehen, folglich die Linie, die den größten Radius aufweist“. Diese Definition der Ideallinie beinhaltet zwei wichtige Unterschiede: Zum einen ist sie nur auf den Radius bezogen und somit rein geometrisch und nicht fahrzeugspezifisch. Zum anderen führt die Linie mit dem größten Radius i. A. nicht zur Zeitoptimalität, wie im weiteren Verlauf gezeigt wird. Darüber hinaus ist die Definition über den „größten Radius“ in Bezug auf einen gesamten Rundkurs nicht eindeutig: Denkbar wäre, das Integral über die quadrierte Krümmung entlang der gesamten Strecke zu minimieren. Linien, die mit einem damit vergleichbaren, geometrischen Gütemaß konstruiert werden, werden im Folgenden *krümmungsminimale Linie* genannt.

Im Rennsport gibt es noch weitere bekannte Fahrlinien, wie die *Kampflinie*, bei welcher der Fahrer mit einem Geschwindigkeitsüberschuss in die Kurve hineinfährt, spät bremst und einlenkt, um im Kurveneingang nicht überholt werden zu können. Durch das späte Einlenken ergibt sich jedoch ein engerer Radius, sodass die Kurve insgesamt langsamer durchfahren werden muss. Da der Hinterherfahrende auch abbremsen muss, ist ein Überholen auch am Kurvenausgang kaum möglich. Für die vorliegende Arbeit haben solche Linien wenig Relevanz. In [88, Kap. 2.1] befindet sich eine kurze Beschreibung verschiedener Linien.

### Bahn und Trajektorie

Genau genommen beschreibt eine Ideallinie lediglich eine *Bahn* oder auch Kurve im mathematischen Sinn. Sie enthält keine Zeitinformation, oder anders formuliert kein Geschwindigkeitsprofil, und erst recht keine Informationen über den Verlauf der Systemzustände. Die Änderung der Zustände im Zustandsraum in Abhängigkeit der Zeit (oder eines anderen kontinuierlich fortschreitenden Parameters) beschreibt eine *Trajektorie*, s. [29, S. 244]. Dennoch wird der Begriff Ideallinie oft synonym zu Trajektorie verwendet, d. h. man geht davon aus, dass zumindest das dazugehörige Geschwindigkeitsprofil bekannt ist. Ebenso wird Bahnplanung im allgemeinen Sprachgebrauch oft synonym für Trajektorienplanung verwendet.

### Definition der Referenzlinie

Die *Referenzlinie* oder kurz *Referenz* stellt keine Fahrlinie dar, sondern ist die mathematische Beschreibung des Streckenverlaufs. Sie verläuft entlang der Fahrbahn, also weist überall einen spitzen Winkel zum Fahrbahnverlauf auf. Es ist naheliegend, sie ungefähr in die Fahrbahnmitte zu legen, sie könnte aber auch sogar jenseits vom Fahrbahnrand liegen oder die Fahrbahn im spitzen Winkel kreuzen. Der örtliche Verlauf der Fahrlinie wird nur durch lokale Größen bezogen auf die

Referenz beschrieben. Für eine Umrechnung in globale Koordinaten ist die Referenz notwendig. In dieser Arbeit werden viele Größen in Abhängigkeit des Weges  $s$  angegeben. Dabei wird die Fzg-Position immer orthogonal auf die Referenz projiziert.  $s$  ist die Bogenlänge der Referenzlinie bis zu diesem Punkt und nicht die tatsächlich zurückgelegte Wegstrecke des Fahrzeugs. Die Referenzlinie kann man auch als FRENET-Pfad bezeichnen.

### Offline- und zyklische Online-Planung

In dieser Arbeit gibt es zwei Modi der Trajektorienplanung: Die *Offline*- und die *zyklische Online*- oder *Echtzeit-Planung*. Die Offline-Planung dient dazu, vor Fahrtbeginn eine Ideallinie (Trajektorie) zu berechnen. In der Regel wird hierbei der gesamte Rundkurs ohne Berücksichtigung anderer Verkehrsteilnehmer geplant. Die Online-Planung wird während der Fahrt zyklisch gestartet und berechnet eine Anschluss trajektorie für den nächsten Streckenabschnitt unter Berücksichtigung etwaiger Objekte, das meint allgemein Hindernisse auf der Strecke.

### Notation des Zählindex

Wenn Variablen an diskreten Stellen angegeben werden, z. B. Wegstellen entlang der Referenz, erhalten sie einen fortlaufenden Index. Hierbei ist der Zählindex *einsbasiert*, d. h. der erste Diskretisierungspunkt heißt  $s_1$  und nicht  $s_0$ . Auch wenn dies auf den ersten Blick ungewöhnlich erscheint, bietet es in der Notation den Vorteil, dass sich bei einer Diskretisierung  $s = s_1, \dots, s_N$  gerade  $N$  Diskretisierungspunkte ergeben. Außerdem entspricht die einsbasierte Notation der Konvention in FORTRAN, einer Programmiersprache, die für numerische Berechnungen in Wissenschaft und Technik verbreitet ist und auch die Basis für MATLAB bildet.

### Notation von Gradient, JACOBI-Matrix und HESSE-Matrix

In dieser Arbeit werden sowohl der stehende als auch der liegende Gradient häufig verwendet. Es gilt folgende Konvention:

$$\nabla := \begin{bmatrix} \frac{\partial}{\partial x_1} \\ \vdots \\ \frac{\partial}{\partial x_n} \end{bmatrix}, \quad \nabla^\top := \begin{bmatrix} \frac{\partial}{\partial x_1} & \cdots & \frac{\partial}{\partial x_n} \end{bmatrix} \quad (1.1)$$

Daraus ergibt sich die JACOBI-Matrix einer vektorwertigen Funktion  $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$

$$\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} = \nabla^\top \mathbf{f}(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial f_1(\mathbf{x})}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_m(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial f_m(\mathbf{x})}{\partial x_n} \end{bmatrix} \in \mathbb{R}^{m \times n}. \quad (1.2)$$

Mit der Definition  $\mathbf{J}_f := \nabla^\top \mathbf{f}$  ist  $\mathbf{J}_f^\top := \nabla \mathbf{f}^\top$  und somit ist  $\nabla^\top \mathbf{f} \neq \nabla \mathbf{f}^\top$ .

Die HESSE-Matrix einer Funktion  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  ist dann

$$\begin{bmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_n} \\ \vdots & & \vdots \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_n^2} \end{bmatrix} = \frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x}^2} =: \nabla^2 f(\mathbf{x}) = \frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^\top} = \nabla^\top (\nabla f(\mathbf{x})). \quad (1.3)$$

## 1.4 Gliederung der Arbeit

Im Anschluss an dieses einführende Kapitel legt **Kapitel 2** zum einen allgemein konzeptionelle Unterschiede vorhandener Ansätze zur Trajektorienplanung und zum anderen konkret das Konzept dieser Planung dar. Die Konzepte werden nach grundlegenden Kriterien kategorisiert. Der Literaturüberblick mit einem Vergleich zur vorliegenden Arbeit folgt am Ende des Kapitels, um darin auf die zuvor beschriebenen Kriterien Bezug nehmen zu können.

**Kapitel 3** beschreibt die Grundlagen dieser zeitoptimalen Trajektorienplanung, i. e. das Koordinatensystem bzw. die Referenzlinie, das primäre Optimierungsziel und das Fahrzeugmodell, welche zu einem nichtlinearen Optimierungsproblem führen. Einzelne Aspekte der Modellierung der Fahrzeugdynamik werden simulativ untersucht.

**Kapitel 4** behandelt die mathematischen Grundlagen und ausgewählte Lösungsmethoden der nichtlinearen Optimierung. Es ist zwar weitgehend losgelöst von dem konkreten Anwendungsfall, fokussiert sich aber stark auf die für diese Arbeit relevanten Themen. Am Ende des Kapitels wird speziell auf den hier verwendeten Optimierungsalgorithmus IPOPT eingegangen.

In **Kapitel 5** wird aus dem Systemmodell und dem Optimierungsziel aus Kapitel 3 ein Optimierungsproblem so formuliert, dass es gut auf die Methoden aus Kapitel 4 abgestimmt ist. Dabei spielt die Rechenzeit eine wesentliche Rolle. Sowohl das Gütemaß als auch das Fahrzeugmodell werden um zusätzliche Komponenten erweitert. Bei der anschließenden Implementierung (Abschnitt 5.4) stehen wiederum Effizienz und Rechenzeit im Vordergrund. Danach präsentiert Abschnitt 5.5 Simulationsergebnisse der Trajektorienplanung für einen Rundkurs. Abschnitt 5.6 beinhaltet einen Exkurs zu einem Ansatz, mit dem durch einen zusätzlichen Güte-term komfortable Trajektorien für den normalen Straßenverkehr berechnet werden können.

**Kapitel 6** beschreibt weitere wesentliche Elemente der Toolchain für die Trajektorienplanung. Sie beginnt mit der Erstellung einer geeigneten Referenzlinie aus gemessenen Fahrbahnrandern. Dabei kommt ein eigens dafür entwickelter Algorithmus zum Einsatz, der die Referenzlinie mittels quadratischer Optimierung glättet (Abschnitt 6.2). Er kann allerdings nicht nur als Filter eingesetzt werden, sondern auch zur Berechnung krümmungsminimaler Rennlinien, was einen Vergleich sowohl mit der zeitoptimalen Trajektorienplanung als auch mit entsprechenden Ansätzen aus der Literatur anbietet. Anschließend beschreibt Abschnitt 6.3 einen sogenannten Matching-Algorithmus zur Referenzierung eines Punkts auf einem Spline.

Nachdem die gesamte Planung zur Offline-Berechnung vorliegt, erweitert **Kapitel 7** sie zu einer zyklischen Echtzeit-Planung. Dabei werden sich ändernde Umgebungsbedingungen in Form von Objekten berücksichtigt. Das Kapitel bespricht Simulationsergebnisse sowohl zur zeitoptimalen als auch zur Komfort-Planung, gibt statistische Auswertungen der Rechenzeiten an und analysiert Messergebnisse einer realen Erprobungsfahrt.

**Kapitel 8** schließt die Arbeit mit einer Zusammenfassung und einem Ausblick auf mögliche weitere Forschungsthemen ab.

## 2 Konzeption der Trajektorienplanung

Sowohl für die Trajektorienplanung im normalen Straßenverkehr als auch für die Berechnung von Ideallinien gibt es viele Ansätze. Drei wesentliche Kriterien, nach denen man die zugrundeliegenden Konzepte kategorisieren kann, sind im Folgenden beschrieben:

- Wird der Lösungsraum durch Vorabwissen auf eine endliche Menge an Trajektorien oder ein vorgegebenes Bewegungsmuster eingeschränkt?
- Erfolgt eine gemeinsame oder separate Quer- und Längsplanung?
- Sind die physikalischen Größen zeit- oder wegababhängig parametrisiert?

Die folgenden drei Abschnitte beschreiben für jedes dieser drei Kriterien die wichtigsten Vor- und Nachteile zusammen mit Beispielen aus der Literatur. Gleichzeitig wird der Ansatz dieser Arbeit anhand der aufgestellten Kriterien eingeordnet.

Darüber hinaus ist auch die Modellierung der Fahrzeugdynamik ein wichtiges Kriterium. Die Variantenvielfalt dafür ist riesig. Sie reicht von einfachen linearen Massepunkt-Modellen und quasi-statischen (QSS) Approximationen bis hin zu komplexen dynamischen Modellen mit unterschiedlicher Anzahl an Freiheitsgraden (DoF). Ein anwendungsorientierter Kompromiss zwischen Modellgenauigkeit und Rechenaufwand ist hierbei sehr wichtig. Deswegen wird an dieser Stelle darauf verzichtet, Ansätze aus der Literatur pauschal in Modellklassen einzuteilen. Stattdessen wird im Literaturüberblick zur Rundenzeitoptimierung in Abschnitt 2.4 auch auf die Art der Modellierung eingegangen.

### 2.1 Definition des Lösungsraums

Idealerweise wird jede – physikalisch mögliche – Trajektorie aus einem kontinuierlichen Lösungsraum als potentielle Lösung zugelassen. Wird die Trajektorienplanung anhand einer mathematischen Optimierung durchgeführt, ist das Ergebnis kein einzelner Wert aus dem Euklidischen Raum  $\mathbb{R}^n$  (*Parameteroptimierung*), sondern eine Funktion von einer unabhängigen Variable, meist von der Zeit. Somit ist das Optimierungskriterium nicht nur eine Gütefunktion, sondern ein Gütefunktional, und die entsprechende Methodenlehre bezeichnet man als *dynamische Optimierung* [60, S. 185]. Die Beschreibung und Lösung des Optimierungsproblems ist nur bei sehr einfachen Problemen analytisch möglich, meist erfolgt sie numerisch. Da sich die numerische Optimierung (zeit-)kontinuierlicher Systeme deutlich komplexer gestaltet, werden meist diskrete oder diskretisierte Modelle verwendet [1, S. 438]. Dient diese Diskretisierung lediglich der numerischen Lösung und ist daher deutlich feiner als die zu untersuchende Systemdynamik, kann der Lösungsraum immer noch als quasi-kontinuierlich angesehen werden.

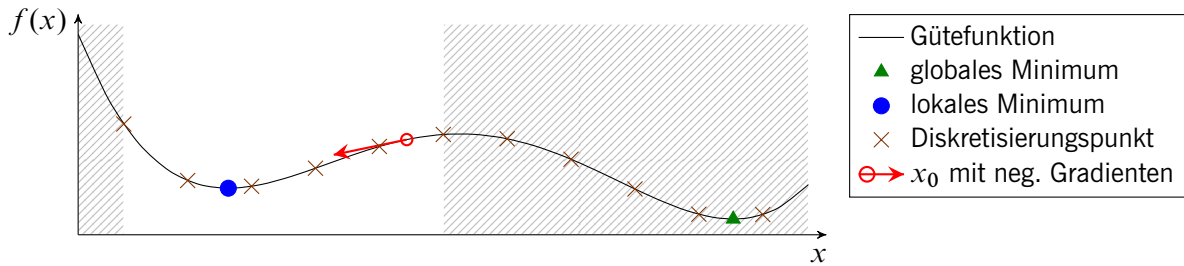
Da die Implementierung der entsprechenden Algorithmen i. d. R. auf Digitalrechnern erfolgt, welche digital und somit wertdiskret arbeiten, entsteht auch eine Quantisierung durch die Darstellung der Gleitkommazahlen, auf die in Anhang A eingegangen wird. Sie ist jedoch bei sinnvoller Problembeschreibung für die Definition des Lösungsraums vernachlässigbar gering.

Die anfangs beschriebene Systemdiskretisierung ermöglicht eine Aussage über die Systemzustände zu den Abtastpunkten. Für den ebenfalls abgetasteten Stellgrößenverlauf wird über eine Ansatzfunktion festgelegt, wie er sich zwischen den Abtastpunkten verhält. Im einfachsten Fall setzt man eine Treppenfunktion an und approximiert die Integration der System-DGLs mit einem expliziten Euler-Verfahren. CASANOVA [20] verwendet zudem für den Stellgrößenverlauf in der Optimalsteuerung eine gröbere Diskretisierung als für die Integrationsschrittweite in der Simulation des Modells und interpoliert die Stellgrößen an den so entstehenden Zwischenstellen [20, S. 116, 175], um die Dimension des Optimierungsproblems zu reduzieren.

Um die Dimension des Optimierungsproblems weiter zu reduzieren, können auch Ansatzfunktionen für Ausgangsgrößen des Systems wie Position oder Geschwindigkeit festgelegt werden, welche so gewählt sein müssen, dass sich realistische Trajektorien ergeben. HANSEN [35] setzt bei einem Fahrstreifenwechsel für den Verlauf der Längs- und Querposition des Fahrzeugs je einen kubischen Spline parametrisiert über der Zeit an. Unter der Annahme eines Massepunktmodells lassen sich über den differentiellen Zusammenhang zwischen Position, Geschwindigkeit, Beschleunigung und Ruck alle diese Größen allein in Abhängigkeit des abschnittsweise konstanten Rucks und den entsprechenden Anfangswerten angeben [35, Kap. 5.2]. Durch die Einschränkung des Lösungsraums auf kubische Polynome kann man das gesamte Fzg-Verhalten analytisch in Abhängigkeit weniger Parameter beschreiben, was zu einer effizienten Planung führt. Diejenigen Terme im Gütemaß, die sich nicht einfach analytisch auf den Ruck zurückführen lassen, werden auf Basis von Abtastpunkten der Splines berechnet [35, S. 55].

GADOLA ET AL. [31] nutzen kubische Splines in einem Tool zur Rundenzeitsimulation, welches die Position von 4 charakteristischen Punkten, nämlich Brems-, Einlenk-, Beschleunigungs- und Kurvenausgangspunkt in Abhängigkeit gewisser Fzg-Parameter optimiert. Der Grund für die Wahl kubischer Splines liegt darin, dass sie die geringste Krümmung innerhalb der Spline-Familie haben [31], was für eine Ideallinie prinzipiell vorteilhaft ist, wie in Abschnitt 2.2 gezeigt wird.

Kubische natürliche Splines sind 2-mal stetig differenzierbar. Ihre Krümmung ergibt sich aus ihrer 2. Ableitung. Somit ist die Krümmung eines kubischen natürlichen Splines ein Spline 1. Ordnung, also eine stückweise affine Funktion, die stetig *und* differenzierbar aber nicht stetig differenzierbar ist. Wie auch aus dem Straßenbau bekannt ist, lassen sich Fahrlinien durch Klothoiden (Kurven, die sich aus der umgekehrten Proportionalität ihres Krümmungsradius zur Länge des Bogens ergeben [16, S. 107]) beschreiben. Daher werden kubische natürliche Splines oft eingesetzt, um Klothoiden zu approximieren. In diesem Projekt hat sich jedoch gezeigt, dass Splines zwar sehr gut geeignet sind, um die Linie (Bahn) zu beschreiben, jedoch führt die fehlende stetige Differenzierbarkeit der Krümmung zu einem unruhigen Verlauf in den fahrdynamischen Größen Gier- und Schwimmwinkel (vgl. (5.28) auf S. 79). Dies führt zu einem unkomfortablen Verhalten, das sich im fahrdynamischen Grenzbereich besonders drastisch auswirkt. Um dies auszugleichen, müssten



**Bild 2.1:** Bei einer Diskretisierung des Lösungsraums wird die Gütefunktion an einzelnen Stellen ausgewertet. Gradientenverfahren laufen vom Startpunkt in ein lokales Minimum.

die Stützstellen des Splines recht eng gewählt werden, wodurch sich wiederum die beiden Vorteile des geringen Rechenaufwands und der glatten (oszillationsfreien) Approximation relativierten.

Während die bisherigen Methoden von einem quasi-kontinuierlichen Lösungsraum ausgehen, beschreibt WERLING in [89, S. 31 ff.] die zu planende Trajektorie in Längs- und Querrichtung durch je ein Polynom 5. Ordnung, dessen Endpunkte durch die Wahl von Endzeit und -position in diskreten Schritten variiert werden. Im Gegensatz zu [35] führt diese diskrete Zielmannigfaltigkeit zu einer Trajektorienschar, aus welcher die beste (ruckminimale) Trajektorie, welche alle Beschränkungen einhält, ausgewählt wird. Ist dabei keine Beschränkung aktiv, ist zeitliche Konsistenz aufgrund des *Bellmanprinzips* gewährleistet [89, S. 27], d. h. bei einer zyklischen Neuplanung ergibt sich auf dem entsprechenden Abschnitt dieselbe Lösung wie zuvor. Die Endgeschwindigkeit und -beschleunigung ergeben sich aus einer Referenztrajektorie abhängig von der Endposition. „Zur konkreten Generierung von Längs- und Querbewegungen reicht es [...] aus, die Referenztrajektorie problemorientiert zu wählen“ [89, S. 38], jedoch muss dafür auch eine irgendwie geartete Taktikplanung vorgeschaltet werden. Eine vergleichbare Diskretisierung beschreibt auch RATHGEBER in [64]: Für die lokale Planung wird eine Trajektorienschar mit Polynomen 7. Ordnung verwendet, die auf einer diskreten Menge von Zuständen endet. Für jede Kombination aus Quer- und Längsplanung wird das Gütemaß bestimmt und die beste Trajektorie unter Einhaltung der Randbedingungen ausgewählt [64, S. 69].

Die unterschiedlichen Ansätze der Diskretisierung des Lösungsraums sollen anhand einer einfachen Parameteroptimierung mit nur einer Optimierungsvariablen  $x \in \mathbb{R}$  mit Bild 2.1 veranschaulicht werden. Die Gütefunktion  $f(x)$  habe ein lokales und ein globales Minimum. Bei der Diskretisierung des Lösungsraums bzw. der Zielmannigfaltigkeit wird  $f(x)$  an Diskretisierungspunkten (DP) ausgewertet und der beste Wert bildet das Optimum. Natürlich müssen die DP nicht wie in Bild 2.1 äquidistant verteilt sein. Bezüglich [89] ist jeder DP vergleichbar mit einer Trajektorie aus der gesamten Schar. Wird die Trajektorie nun aber nicht als Spline mit wenigen Parametern angegeben, sondern im Zustandsraum beschrieben, steigt die Anzahl aller Kombinationen exponentiell mit der Dimension des Zustandsraums. Wollte man dafür auch alle Trajektorien auswerten und die beste, die alle Nebenbedingungen einhält, auswählen, würde die Rechenzeit exorbitant steigen, was BELLMAN als „Fluch der Dimensionalität“ bezeichnet [7, Kap. 5.16]. Die Dynamische Programmierung reduziert die Auswertungen daher auf diejenigen Kombinationen, die aufgrund der Systemdynamik erreichbar sind [7, Kap. 5.8], was i. A. immer noch zu einer immens hohen



Anzahl führt. Gradientenbasierte Verfahren hingegen starten an einem Punkt  $x_0$  (vgl. Bild 2.1), werten an der Stelle den Gradienten aus und laufen eine gewisse Schrittweite in Richtung des negativen Gradienten. Das wird so lange wiederholt, bis ein lokales Minimum gefunden wird.

Vorteil bei einer Trajektorienschär ist die hohe Wahrscheinlichkeit, das globale Optimum zu finden, auch wenn die Gütefunktion mehrere Minima hat. Nachteilig ist, dass viele Funktionsauswertungen an Stellen durchgeführt werden müssen, die weit vom Optimum entfernt sind, während gradientenbasierte Verfahren in einem kontinuierlichen Lösungsraum gezielt in Richtung des (lokalen) Optimums laufen. Die Anzahl der Schritte hängt dann von der Qualität der Startlösung und der Liniensuche (Schrittweitensteuerung) sowie den Eigenschaften der Gütefunktion ab. Die Berechnung des Gradienten ist zwar meist weit aufwendiger als eine Auswertung der Gütefunktion, jedoch ist die Anzahl der Funktionsauswertungen beim diskreten Ansatz i. d. R. viel höher. Bei beiden Ansätzen ist es ggf. notwendig, den Lösungsraum einzuschränken – beim diskreten um die Dimension zu reduzieren und beim kontinuierlichen um unerwünschte lokale Optima auszuschließen, wie es bspw. in Bild 2.1 durch den nicht schraffierten Bereich dargestellt ist. In [89, 64] wird der (diskrete) Lösungsraum durch die Wahl der Referenztrajektorie bzw. des Endzustands beschränkt, was einer vorgeschalteten Taktik-/Manöverplanung entspricht. Eine zweistufige Planung kann zudem auch bei einer Trajektorienschär dazu genutzt werden, den Lösungsraum auf ein spezielles globales oder auch lokales Minimum zu beschränken. Zur Berechnung einer Ideallinie ist die Verwendung einer Trajektorienschär aber wenig zielführend, weil Ideallinien aus zu vielen unterschiedlichen Mustern bestehen, als dass man sie mit einer sinnvollen Anzahl an Trajektorien angeben könnte, wodurch der Lösungsraum zu groß würde.

In dieser Arbeit wird eine Diskretisierung des Optimierungsproblems lediglich zur numerischen Integration durchgeführt. Die Modellgleichungen werden über Nebenbedingungen im Optimierungsproblem berücksichtigt und das resultierende Optimierungsproblem mit IPOPT [86] gelöst. Als Ansatzfunktion für die Stellgrößen dienen Treppenfunktionen, da sie zu einer möglichst dünnbesetzten JACOBI-Matrix führen. Somit ermittelt die Planung nur lokale Optima und kann nicht ohne Weiteres als Taktikplanung verwendet werden. Um den Lösungsraum auf einen Bereich um ein gewünschtes Optimum einzuschränken, kann eine (semi-)globale Taktikplanung vorgeschaltet werden, die bspw. vorgibt, welcher Fahrstreifen zu verwenden ist oder ob und wie ein anderes Fahrzeug überholt werden soll, was für einen späteren Einsatz im normalen Straßenverkehr notwendig ist. Für eine Taktikplanung können kombinatorische Ansätze evtl. auf Basis einfacher kinematischer Zusammenhänge zum Einsatz kommen. In [64] wird für die (semi-)globale Planung ein Verfahren vorgestellt, das für jeden befahrbaren Fahrstreifen unter Berücksichtigung einfacher Abstandsgesetze eine Zieltrajektorie berechnet. Dabei wird der Lösungsraum in Querrichtung stark diskretisiert, indem nur die Mitte jedes Fahrstreifens als Lösung betrachtet wird. Über ein Gütemaß wird die beste Trajektorie ausgewählt, welche die maximal erlaubten Längs- und Querschleunigungen einhält, und so der Zielbereich für die Feinplanung vorgegeben.

Eine solche zweistufige Planung kann gegenüber einer einzigen globalen Gesamtplanung Rechenzeit sparen, weil globale Optimierungsverfahren, wie z. B. die Dynamische Programmierung, bei hoher Modellgenauigkeit und feiner Diskretisierung mitunter extrem lange Rechenzeiten erfordern. Einzige Gefahr bei diesem Vorgehen ist, dass die anschließende Feinplanung möglicher-

weise gar keine Lösung findet, weil die Grobplanung zu optimistisch war. Um das zu umgehen, kann die Grobplanung einen sicheren Alternativzielbereich vorsehen, für den eine Lösung existiert (z. B. „wenn möglich Fahrstreifenwechsel, anderenfalls abbremesen“). Es verdoppelt natürlich die Rechenzeit für die Feinplanung, wenn im Worst Case zwei Lösungen zu berechnen sind.

Eine andere Möglichkeit, den Lösungsraum einzuschränken, ergibt sich aus der Annahme, dass das Optimum in der Nähe einer Referenztrajektorie liegt. Voraussetzung hierfür ist natürlich die Kenntnis über diese Referenztrajektorie, die bspw. aus einer Vorausplanung oder – im Falle von Rennlinienberechnungen für vorgegebene Rennstrecken – aus einer manuell eingefahrenen Trajektorie stammen könnte. Dann kann das Problem um diese Trajektorie herum linearisiert werden, wie es TIMINGS UND COLE [81] zur Berechnung von Ideallinien mittels konvexer Optimierung vorgeschlagen haben. Problematisch daran ist jedoch, dass die Dynamik im Grenzbereich so nichtlinear ist, dass schon kleine Abweichungen den Gültigkeitsbereich der Linearisierung verlassen. Bei Untersuchungen im Rahmen dieser Arbeit hat sich gezeigt, dass der rechentechnische Aufwand, allein sicherzustellen, dass die resultierende Trajektorie bezogen auf das nichtlineare Modell fahrbar ist, zu groß ist, um einen Vorteil gegenüber der hier verwendeten nichtlinearen Optimierung zu erreichen [102].

Eine im Motorsport verbreitete Vorausplanung ist die Strategieplanung. Sie gibt bspw. an, wann Boxenstopps eingelegt werden und welcher Reifentyp in welchem Stint (Zeit, die ein Rennfahrer im Rennen fährt, ohne durch einen Boxenstopp zu unterbrechen) zum Einsatz kommt, wie es in [36] beschrieben ist. Auf eine Strategieplanung wird in dieser Arbeit aber nicht eingegangen.

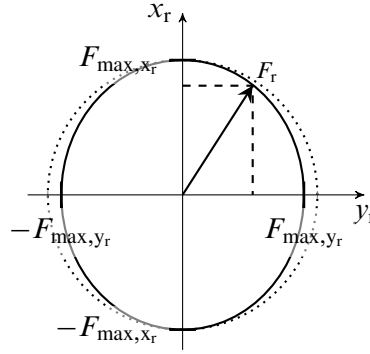
## 2.2 Wechselwirkung von Längs- und Querplanung

In der vorliegenden Arbeit werden Querplanung (Bahnplanung) und Längsplanung (Geschwindigkeitsprofil) in einem Schritt gemeinsam berechnet, um die optimale Lösung zu erhalten. In anderen Anwendungsfällen können sich aber auch hinreichend gute Lösungen ergeben, wenn zuerst eine Bahn berechnet und anschließend daraus ein Geschwindigkeitsprofil ermittelt wird. Um diese beiden grundsätzlichen Möglichkeiten gegenüberstellen zu können, ist es sinnvoll, zunächst ein Vorgehen zur getrennten Längs- und Querplanung in den Grundzügen zu erläutern.

Eine solche Bahnplanung kann über eine Minimierung der quadrierten Krümmung  $\kappa$  erfolgen, also einer rein geometrischen Größe, sodass die Linienwahl unabhängig von der Fahrzeugdynamik ist. Zum Bsp. minimieren BRAGHIN ET AL. [12] eine Kombination aus Krümmung und Streckenlänge. Denn mit der Zentripetalbeschleunigung  $a_c$ , welche durch den Kraftschlussbeiwert  $\mu$  begrenzt ist, lässt sich gemäß

$$a_c = \kappa v^2 \quad \text{mit} \quad |a_c| \leq \mu g \quad (2.1)$$

die Geschwindigkeit  $v$  maximieren, wenn der Betrag der Krümmung minimal ist. Mit (2.1) folgt für jedes Streckensegment mit  $\kappa \neq 0$  eine obere Grenze für die Geschwindigkeit und insbesondere die maximale Geschwindigkeit in jedem Kurven-Apex (Kurvenscheitel, Maximum von  $|\kappa|$ ). Um ein realistisches Geschwindigkeitsprofil zu berechnen, muss das Beschleunigungs- und Verzögerungspotential des Fahrzeugs eingehalten werden. Neben der Motorleistung ist der Kraftschluss



**Bild 2.2:** Der Kammsche Kreis beschreibt die Grenzkurve der übertragbaren Horizontalkraft am Reifen. Reale Reifen weisen eine leicht elliptische Grenzkurve auf.

zwischen Reifen und Fahrbahn entscheidend dafür. Der Betrag der maximal übertragbaren Horizontalkraft  $F_r$  zwischen Fahrbahn und Reifen ist bei COULOMBScher Reibung auf

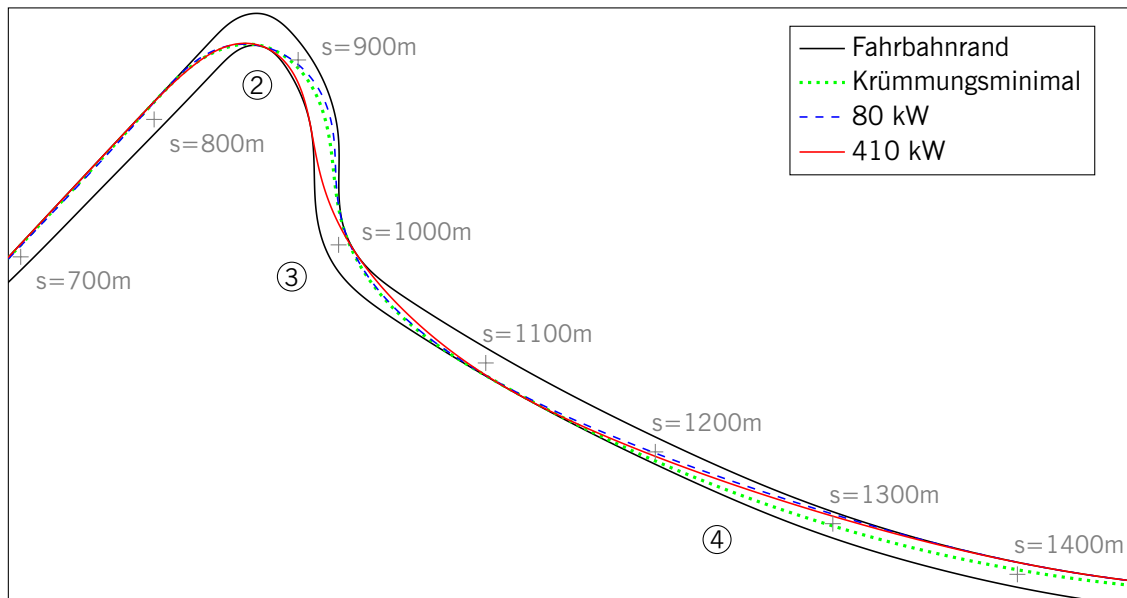
$$|F_r| = \sqrt{F_{x_r}^2 + F_{y_r}^2} \leq \mu F_{z_r} \quad (2.2)$$

begrenzt, wobei  $F_{z_r}$  die Normalkraft ist. Stellt man diesen Zusammenhang in einem Diagramm dar, ergibt sich ein Kreis, weshalb der Zusammenhang in der Fahrdynamik *Kammscher Kreis* genannt wird [15, S. 24]. In der Realität hat ein Reifen u. a. durch die Profile in Längs- ( $x_R$ )- und Quer- ( $y_R$ )-Richtung etwas unterschiedliche Eigenschaften. So ist die Grenzkurve der maximal übertragbaren Kraft eine Ellipse, da bei realen Reifen der Kraftschlussbeiwert in Längsrichtung  $\mu_x$  i. d. R. etwas größer ist als in Querrichtung  $\mu_y$  [75, S. 167]. Ebenso gibt es kleine Unterschiede für positive und negative Längskräfte, die aber hier vernachlässigt werden. So ergibt sich aus (2.2):

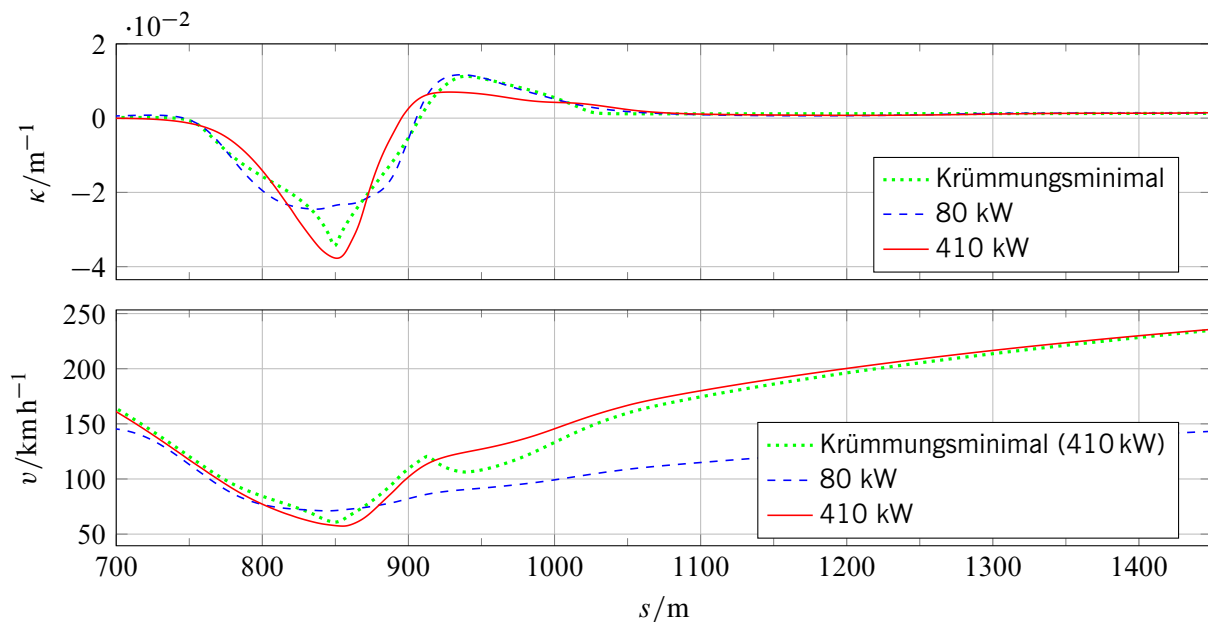
$$\sqrt{\frac{F_{x_r}^2}{F_{\max, x_r}^2} + \frac{F_{y_r}^2}{F_{\max, y_r}^2}} \leq 1, \quad F_{\max, x_r} := \mu_x F_z, \quad F_{\max, y_r} := \mu_y F_z. \quad (2.3)$$

Dieser Zusammenhang ist in Bild 2.2 als durchgezogene Linie dargestellt.

Ausgehend von einer Punktmasse kann das Verhalten mit dem NEWTONschen Gesetz  $F = ma$  auf die Schwerpunktsbeschleunigung übertragen werden und die Zentripetalbeschleunigung  $a_c$  ist gleich der Querbeschleunigung  $a_y$ . Üblicherweise hat das Fzg im Apex eine reine Querbeschleunigung, da man dort weder verzögert noch beschleunigt. Ausgehend von den Apizes, an denen die Geschwindigkeit also lokal minimal ist, kann der Geschwindigkeitsverlauf nach jedem Apex durch Vorwärts-Integration der maximal möglichen Längsbeschleunigung berechnet werden und der Geschwindigkeitsverlauf vor jedem Apex durch Rückwärts-Integration der maximal möglichen Verzögerung. Das Umschalten von Beschleunigen auf Verzögern erfolgt jeweils an der Stelle, an der beide simulierten Geschwindigkeiten identisch sind. Derartige Verfahren werden u. a. in [12, 48, 14, 70] eingesetzt. Auch im Rahmen dieser Arbeit wurde es zu Vergleichszwecken implementiert [104]. Problematisch ist es, dass durch die Umschaltung Knicke im Geschwindigkeitsprofil entstehen. Daher ist eine Ruckbegrenzung oder Glättung notwendig. Vorteil des Verfahrens ist die kurze Rechenzeit, die für eine Strecke von 5 km bei nur einigen Millisekunden liegt.



(a) Linienführung für einen Abschnitt des Hockenheimrings [95]



(b) Krümmungs- und Geschwindigkeitsverlauf der einzelnen Linien

**Bild 2.3:** Krümmungsminimale Linie im Vergleich zu zeitoptimalen Linien für Beispielfahrzeuge mit 80 kW und 410 kW Motorleistung

Da geometrisch berechnete (krümmungsminimale) Linien selten zeitoptimal sind, können Quer- und Längsplanung iterativ wiederholt werden. So berechnet KAPANIA [48] für die ermittelte Geschwindigkeit der Ausgangslinie eine neue Linie durch Minimierung einer Kombination aus Gier- und Lenkrate. Daraus lässt sich dann wieder ein neues Geschwindigkeitsprofil berechnen.

An einem Streckenabschnitt des *Hockenheimrings* von der „Bernie Ecclestone-Kurve“ (Kurve 2) bis zur „Parabolika“ (Kurve 4) in Bild 2.3(a) kann man gut erkennen, dass die Ideallinie stark von Fahrzeugparametern beeinflusst wird: Grün gepunktet ist eine näherungsweise krümmungs-

minimale Linie dargestellt, welche mit dem in Abschnitt 6.2 beschriebenen Algorithmus erstellt wurde und nur geometrische Größen berücksichtigt. Die blau gestrichelte und die rot durchgezogene Linie wurden als zeitoptimale Trajektorie für ein seriennahes Fahrzeug mit einer (fiktiven) Motorleistung von 80 kW (107 PS) bzw. 410 kW (550 PS) und Allradantrieb berechnet. Bild 2.3(b) zeigt den Krümmungsverlauf der Linien sowie das Geschwindigkeitsprofil, wobei die Geschwindigkeit für die krümmungsminimale Linie für eine Motorleistung von 410 kW mittels Vorwärts-/Rückwärts-Integration berechnet wurde.

Ein Rennfahrer schaut immer zuerst auf die schnellen Passagen einer Strecke und überlegt sich dann, wie er am Ausgang der letzten Kurve vor dieser Passage fahren muss, um eine gute Ausgangsposition für diese Passage zu bekommen. Daraus lässt sich ableiten, wie die Kurve eingangsseitig angefahren werden muss, um eben diese perfekte Position für die schnelle Passage zu erhalten. Er geht also die Strecke mental rückwärts ab. Die zeitoptimale Trajektorienplanung berücksichtigt solche Tatsachen automatisch.

Das hoch motorisierte Fahrzeug fährt Kurve 2 sehr eng und verschafft sich so eine gute Ausgangsposition für den Eingang zur „Parabolika“, indem es zum einen schon früh hohe Beschleunigungen aufbringen kann (ab  $s \geq 860$  m) und zum anderen die Kurve 3 im Bereich  $s \approx 950$  m aufgrund der geringen Krümmung  $|\kappa| < 0,01 \text{ m}^{-1}$  ohne Geschwindigkeitsverlust passieren kann. Dass der Eingang zu Kurve 2 dabei wegen der hohen Krümmung langsam durchfahren werden muss, ist für die Gesamtzeit trotzdem optimal, da sich der Apex nach vorne verschiebt und die Krümmung am Kurvenausgang schnell abnimmt.

Das schwach motorisierte Fahrzeug nimmt hingegen mit geringer Krümmung möglichst viel Geschwindigkeit mit durch Kurve 2, da verlorene Geschwindigkeit nur langsam wieder aufgebaut werden kann. Im Apex ist es so 14 km/h schneller als das hoch motorisierte. Die Krümmung ist anschließend fast in dem gesamten Bereich  $s = 870 \dots 1010$  m deutlich höher. Im Allg. führt eine hohe Krümmung nach (2.1) zu hohen Zentripetalbeschleunigungen, wodurch weniger Kraftschlusspotential für die Längsbeschleunigung zur Verfügung steht. In diesem Fall ist die Längsbeschleunigung aber durch die Motorleistung so begrenzt, dass ein höheres Kraftschlusspotential gar nicht genutzt werden könnte. Daher ist es optimal, wenn der Krümmungsverlauf keine hohen Spitzen aufweist, die lokal niedrige Geschwindigkeiten fordern. Man sagt auch: „Die Kurve wird runder gefahren.“ Zudem hat das Fahrzeug aufgrund der geringeren Geschwindigkeit in der Parabolika mehr Potential zum Lenken und kann so eine (etwas) kürzere Linie wählen.

Andere Fahrzeugparameter, seien es die Kraftverteilung auf Vorder- und Hinterachse oder die Reifenparameter, ergeben wieder andere individuelle Ideallinien. Und dabei sind Effekte, die sich durch unterschiedliche Kraftschlussbeiwerte auf der Strecke ergeben könnten, noch nicht berücksichtigt. Doch ist bereits so schon offensichtlich, dass die krümmungsminimale Linie zwar einzelne Charakteristika beinhaltet, jedoch für keines der beiden Fahrzeuge eine (zeitoptimale) Ideallinie darstellt. Um die Qualität zu verbessern, kann bspw. wie in [12] zusätzlich die Weglänge berücksichtigt werden, jedoch kann die fahrzeugspezifische ideale Gewichtung nur empirisch ermittelt werden und stellt dennoch keine global optimale Lösung dar. Aus diesem Grund wird hier als Gü-

temaß die echte Fahrzeit berechnet und kein lokales Ersatzmaß angenommen. Die Fahrzeit wird dabei durch eine simultane Optimierung von lateraler und longitudinaler Bewegung minimiert.

## 2.3 Zeit- oder Weg-Parametrierung

Aus dem Alltag ist man gewohnt, dass die Zeit ein unabhängiger Parameter ist, der sich streng monoton verhält. Deswegen werden Bewegungen oft als zeitabhängige Trajektorie beschrieben. Im Straßenverkehr verwendet der Mensch jedoch intuitiv den Ort als Bezugsgröße: Wenn man als Autofahrer eine Kurve vor sich sieht, wird man sich nicht denken: „In fünf Sekunden werde ich für zwei Sekunden das Lenkrad einschlagen“, sondern: „Wenn ich den Kurveneingang erreicht habe, werde ich das Lenkrad einschlagen, bis ich am Kurvenausgang angekommen bin.“ Diese Aussage zeigt anschaulich, dass der Mensch in solchen Situationen über den Ort parametriert und sich die Zeit als Variable in Abhängigkeit der Systemzustände (Geschwindigkeit) ergibt.

Mathematisch wird die Trajektorie dann durch  $\mathbf{x}(s)$  angegeben, wobei  $s$  die Länge einer Referenzlinie entlang des Streckenverlaufs ist (vgl. Definition der Referenzlinie auf S. 7). Die Referenzlinie kann z. B. die Mittellinie der Fahrbahn sein. Sie kann davon aber auch in gewissen Grenzen abweichen, genauso wie sich der Mensch auch nicht an einer exakten Mittellinie orientiert.

Ein Vorteil der Orts-Parametrierung ist besonders relevant: Wenn das System an einem Ort einen gewissen Zustand einnimmt, bleibt dieser Zustand an derselben Stelle der Trajektorie, unabhängig davon, wie schnell sich das System in dem vorhergehenden Trajektorienstück bewegt hat. So können bspw. die Fahrbahnkrümmung und -breite für jeden Punkt  $s$  als eine von den Optimierungsvariablen *unabhängige* Funktion angegeben werden, was für eine zeitabhängige Parametrierung nicht möglich ist. Nachteilig ist hingegen die Bedingung, dass  $s$  streng monoton steigen muss, sodass kein Stillstand möglich ist. Jedoch gilt diese Bedingung bereits bei Verwendung eines klassischen Einspurmodells, sodass Stillstand häufig separat betrachtet werden muss.

Bei Rundkursen muss die Trajektorie periodisch sein und Anfangs- und Endzustand müssen gleich sein. Bei einer Wegparametrierung lässt sich hierfür die einfache Bedingung  $\mathbf{x}(0) \stackrel{!}{=} \mathbf{x}(S_{\text{end}})$  angeben. Bei zeitlicher Parametrierung wäre diese Bedingung deutlich schwieriger, weil der Endzeitpunkt erst nach der Optimierung bekannt ist.

Konsequenterweise werden bei so einem wegbasierten Ansatz nicht nur alle Größen auf die Referenzlinie bezogen, sondern auch die System-DGLs vom Zeit- in den Wegbereich transformiert. Dies bringt zwar einen zusätzlichen mathematischen Aufwand mit sich, jedoch erlaubt es eine elegante Formulierung des zeitoptimalen Gütemaßes, wie in Kapitel 3 beschrieben wird. Die Transformation in den Wegbereich ist jedoch nicht zwingend notwendig, wenn lediglich ein referenzlinienbasiertes Koordinatensystem zum Einsatz kommen soll. Es gibt zahlreiche Ansätze, darunter [90, 81, 34, 63], die Referenzlinien bzw. Frenet-Koordinaten verwenden, und dennoch die Dynamik im Zeitbereich abbilden und berechnen. Nur [20, 61, 50] transformieren die Fahrzeugdynamik wie auch bei diesem Ansatz in den Wegbereich.

## 2.4 Literaturüberblick zur Rundenzeitoptimierung

In den vorherigen Abschnitten dieses Kapitels wurde eine Möglichkeit zur Kategorisierung von Methoden zur Trajektorienplanung anhand dreier Aspekte vorgeschlagen und dabei ausgewählte Literatur sowohl zur Trajektorienplanung im allgemeinen Straßenverkehr als auch zur Rennlinienberechnung angeführt. Der folgende Literaturüberblick legt den Fokus speziell auf Rennlinienberechnung und Rundenzeitoptimierung. Durch seine chronologische Sortierung lässt sich die historische Entwicklung nachvollziehen.

Frühe Ansätze zur Rennlinienberechnung finden sich bspw. 1996 bei GADOLA ET AL. [31] sowie HENDRIKX, MEIJLINK UND KRIENS [38]. In [31] ist die Fahrlinie einzelner Kurven durch markante Stellen (Brems-, Einlenk-, Beschleunigungs- und Kurvenausgangspunkt) beschrieben, deren Position im Sinne der Rundenzeit optimiert wird. [38] berechnet einen zeitoptimal ausgeführten Fahrstreifenwechsel für ein Punktmassemodell mit einem PACEJKA-Reifenmodell.

MÜLLER, MÜHLMEIER UND MARTIGNONI [56] von Audi Sport verwenden 1999 einen evolutionären Algorithmus, um ausgehend von der Mittellinie eine näherungsweise zeitoptimale Linie zu generieren.

Die meisten Forschungsprojekte im Bereich der Rennlinienberechnung haben das Ziel, die Auswirkung von diversen Fzg-Parametern auf die Rundenzeit bei Formel-Rennwagen zu untersuchen. Die entsprechenden Parameter müssen jedoch in einem Modell überhaupt vorhanden sein, um ihre Auswirkung untersuchen zu können. Daher wurde der Fokus vermehrt auf die Modellierung und die Anzahl der Freiheitsgrade (DoF) des Modells gelegt. SIEGLER, DEAKIN UND CROLLA [77] unterteilen 2000 die Modellierung der Fahrzeugdynamik in die 3 Strategien „statisch“, „quasi-statisch“ (QSS) und „transient“. Während statische Modelle keine zeitabhängigen Variablen enthalten, ist die Strecke bei QSS-Modellen in eine Folge von Segmenten mit konstantem Radius unterteilt, in denen das Fzg-Verhalten als statisch angenommen wird [77]. Transiente Simulationen berücksichtigen dynamische Vorgänge sowie den gegenseitigen Einfluss von longitudinalen, lateralen und rotatorischen Bewegungen. Diese prinzipielle Unterteilung wurde in den folgenden Jahren häufig aufgegriffen. Streng genommen ist die Grenze zwischen quasi-statischen und transienten Modellen jedoch fließend, wenn Modelle einen Teil der Dynamik transient und einen anderen Teil (quasi-)statisch abbilden. Auch führt eine sehr feine Segmentierung der Strecke im Grenzfall zu einer exakten Modellierung des transienten Verhaltens. Aufgrund dieser Uneindeutigkeit wurde in den vorhergehenden Abschnitten eine alternative Kategorisierung gewählt. Ein Merkmal vieler QSS-Methoden ist, dass Längs- und Querplanung separat erfolgen, und sich die Geschwindigkeit aus (2.1) ergibt.

CASANOVA [20] präsentiert 2000 eine transiente Optimierung, welche als Bezugssystem eine Referenzlinie verwendet und das Problem in den Ortsbereich transformiert. So kann er die tatsächliche Fahrzeit im Gütemaß minimieren. Der Stellgrößenverlauf wird anhand von Diskretisierungspunkten („control nodes“) im Abstand von 4–20 m mit SNOPT optimiert, während die Simulation des Fzg-Modells mit einer Integrationsschrittweite von 1–10 cm separat erfolgt [20, S. 175 u. 181] (vgl. *direkt sequentielles Verfahren* in Abschnitt 4.1). Zwischen den control nodes

wird der Stellgrößenverlauf linear interpoliert [20, S. 116]. Da in jedem Iterationsschritt der Optimierung und somit auch in der ersten Iteration eine offene Simulation durchgeführt wird, muss die Startlösung bereits einen Stellgrößenverlauf haben, der zum einen das Fzg auf der Strecke hält und zum anderen zu einem fahrdynamisch stabilen Verlauf führt. Denn eine instabile Simulation oder ein Verlassen der Strecke führt zum Abbruch der Optimierung. Daher wird in [20, Kap. 3] eine Methode zur Rekonstruktion einer manuell „eingefahrenen“ Linie vorgestellt, die als Startlösung dienen kann. Zudem wird die Strecke in Segmente von 8–100 m Länge unterteilt, um die Stellgrößen in jedem Segment entkoppelt betrachten zu können [20, S. 111 ff.]. Die Optimierung sorgt an den Segmentgrenzen für einen stetigen Verlauf relevanter Zustandsgrößen.

An den Stellen, an denen der Fzg-Schwerpunkt voraussichtlich in der Nähe des Fahrbahnrandes liegt, wird die Nähe zum Rand durch einen quadratischen Term bestraft, um die Ränder nicht zu verlassen [20, S. 106 f.]. Da diese quadratische Funktion symmetrisch zur Referenzlinie ist, muss die Referenzlinie in der Fahrbahnmitte verlaufen.

Der Fokus von [20] liegt auf der detaillierten Untersuchung der Auswirkung von Fzg-Parametern auf die Rundenzeit bei Formel-Rennwagen. Die Simulation verwendet ein Zweispurmodell mit 7 DoF [20, S. 159 f.]. Aufgrund der Berechnung der Ableitungen im Optimierungsproblem durch *automatisches Differenzieren* (engl. automatic differentiation, AD, siehe [58, Kap. 8.2]) lässt sich das Modell leicht austauschen, solange Eingangs- und Ausgangsgrößen übereinstimmen.

Die Optimierung der Rundenzeit erfolgt auf eine Millisekunde genau [20, S. 178], um den Einfluss von Fzg-Parametern auf die Linienwahl und die Geschwindigkeit detailliert untersuchen zu können. Die Rechenzeit lag für den 5,8 km langen *Suzuka International Racing Course* (JP) auf einer damals recht performanten *Compaq Alpha Workstation* für 4 Konfigurationen zwischen 7 und 10 Stunden [20, S. 183]. Für den 4,7 km langen *Circuit de Barcelona-Catalunya* (ES) konnte auf einer weniger performanten *Sun Workstation* für eine Konfiguration nach 28 Stunden eine Lösung gefunden werden. Die Optimierungen der 3 anderen Konfigurationen wurden nach 60 Stunden abgebrochen. Die berechneten Rundenzeiten sind etwa 5 s ( $\approx 5\%$ ) schneller als die tatsächlich gefahrenen Bestzeiten professioneller Fahrer, was darauf zurückzuführen ist, dass die Fzg-Parameter unter Laborbedingungen ermittelt wurden und daher von den realen (nicht genau bekannten) Parametern abweichen [20, S. 183].

Für viele folgende Arbeiten dient [20] als Maßstab. Doch gerade für die Rundenzeit-Simulation in der Formel 1 stellten die immens langen Rechenzeiten von Stunden und Tagen ein großes Problem dar, wodurch der Trend zu QSS-Methoden ging, welche in der Formel 1 große Verbreitung fanden [61, S. 654].

BRAYSHAW [13, 14] zeigt 2004 eine QSS-Methode, welche die Fahrlinie aus [20] verwendet und für diese Linie ein optimiertes Geschwindigkeitsprofil für verschiedene Fzg-Konfigurationen berechnet. Über die Krümmung der Fahrlinie ergeben sich die Querschleunigung und die Geschwindigkeit im Kurvenapex. Wie in Abschnitt 2.2 beschrieben, kann durch Vorwärts-/Rückwärts-Integration prinzipiell der gesamte Geschwindigkeitsverlauf ermittelt werden. Dabei bewegt man sich immer auf dem *Kammschen Kreis*. (Man spricht auch vom „GG-Diagramm“, wenn die SP-Beschleunigung statt der Reifenkraft aufgetragen ist, weil die Beschleunigung als Vielfaches



der Erdbeschleunigung  $g$  angegeben wird.) In der Realität ist das GG-Diagramm aber besonders bei Rennwagen nicht fix. Diese sind nämlich so konstruiert, dass sie bei hohen Geschwindigkeiten eine große aerodynamische Abtriebskraft haben, wodurch die Maximalbeschleunigung im Gegensatz zur Annahme in (2.1) geschwindigkeitsabhängig ist, sodass wegen  $F_z \geq mg$  auch  $|a| > \mu g$  sein kann. Daher berechnet BRAYSHAW ein „GG speed diagram“, welches für verschiedene Geschwindigkeiten die Kurven der maximalen Längs- und Querbeschleunigungen angibt [13, S. 10 ff.], [14, S. 729]. Jeder Punkt im GG speed diagram stellt die maximale Beschleunigung simuliert für einen stationären Zustand dar. Diese Berechnung nimmt den größten Teil der Rechenzeit in Anspruch. Insgesamt liegt sie für den Barcelona-Circuit bei 16 Minuten [13, S. 58] und somit erwartungsgemäß deutlich unter den Zeiten von [20]. Jedoch ist dabei zu berücksichtigen, dass der Algorithmus keine Berechnung der Linie, sondern nur der Längsdynamik durchführt. Im Vergleich zu [20] ist die resultierende Rundenzeit von [14] 2,2 s ( $\approx 2,7\%$ ) langsamer.

KELLY [49] (2008) baut zwar auf den Ergebnissen von [20] auf, jedoch statt die Fahrzeit zu minimieren, maximiert er die auf die Mittellinie projizierte Fahrstrecke innerhalb einer festen Zeit. Das Fzg-Modell und die Berücksichtigung der Fahrbahnränder in Form einer quadratischen Straffunktion [49, S. 25] sind wie in [20] gewählt. Allerdings entwickelt er eine Methode, um aus einer zufälligen Startlösung einen gültigen Stellgrößenverlauf für die Optimierung zu generieren [49, S. 34 f.], was einen Vorteil gegenüber [20] darstellt. Der Fokus der Arbeit liegt auf Untersuchungen eines Feder-Masse-Dämpfer-Modells für die Vertikaldynamik und eines temperaturabhängigen Reifenmodells. Die Rechenzeit liegt für den 4,4 km langen *Circuito Permanente de Jerez* (ES) und eine Diskretisierung des Stellgrößenverlaufs von 10 m bei 8 Stunden [49, S. 172].

BRAGHIN ET AL. [12] stellen im selben Jahr einen einfacheren (QSS) Ansatz vor, der eine Kombination aus Krümmung und Weglänge minimiert. Er wurde in Abschnitt 2.2 beschrieben.

GERDTS ET AL. [32] (2009) entwickeln eine Trajektorienplanung mittels MPC, welche in einem automatisiert fahrenden (prototypischen) Golf VI der Volkswagen AG Konzernforschung getestet wurde. Sie maximieren in ihrem Gütemaß die zurückgelegte Strecke innerhalb einer fixen Zeitspanne. Da die Zeitspanne kürzer als die Fahrzeit der gesamten zur Verfügung stehenden Strecke sein muss, kann die Methode nur auf eine Moving-Horizon-Strategie (MPC) und nicht eine gesamte Rundkurs-Planung angewendet werden. Anfangs- und Endpunkt des zurückgelegten Streckensegments werden jeweils auf den linken und rechten Rand projiziert und die Summe dieser beiden Distanzen maximiert [32, S. 448 ff.]. Zusätzlich wird das Integral über der quadrierten Lenkrate minimiert. Hierbei entfällt die Berechnung der Mittel- oder Referenzlinie. Als Modell kommt ein Einspurmodell mit PACEJKA-Reifenkennlinie zum Einsatz. Die Optimierung erfolgt über ein SQP-Verfahren. Die Berechnung einer Runde für die 3,7 km lange *Motorsport Arena Oschersleben* (DE) benötigte bei einem Planungshorizont von 10 s knapp 12 Minuten [32, S. 452].

TIMINGS UND COLE [81] (2013) maximieren wie [49] die gefahrene Strecke projiziert auf die Mittellinie. Jedoch formulieren sie daraus ein quadratisches Optimierungsproblem (QP), indem sie die Fzg-Dynamik um eine Referenztrajektorie herum linearisieren und die Projektion der gefahrenen Strecke in eine TAYLOR-Reihe 2. Ordnung entwickeln. Dieses zeitvariante QP wird im

Rahmen einer MPC gelöst. Ergebnisse werden in [81] für eine  $90^\circ$ -Kurve präsentiert. Die Berechnung eines Horizonts von 20 ms mit 500 Diskretisierungspunkten dauert 900 ms.

PERANTONI UND LIMEBEER [61] (2014) verwenden wie [20] eine Transformation in den Ortsbereich, um für Formel 1-Wagen zeitoptimale Trajektorien für Rundkurse zu berechnen. Jedoch wird das Optimierungsproblem mit der MATLAB-Toolbox ICLOCS diskretisiert und über ein *direkt simultanes Verfahren* (vgl. Abschnitt 4.1) mit einer Trapez-Integration in IPOPT gelöst [61, S. 664]. Im selben Problem können gleichzeitig gewisse Fzg-Parameter optimiert werden. In [61] ist die Diskretisierung konstant 2 m. Das verwendete Modell hat 3 DoF. Es beschreibt die Fzg-Bewegung in kartesischen Koordinaten (äquivalent zu ihrer Formulierung der Weg-/Zeit-Transformation). Zusätzlich berücksichtigt das Modell die Radaufstandskräfte an allen vier Rädern. Nicht stetig differenzierbare Funktionen  $\min(\cdot)$  und  $\max(\cdot)$  werden durch stetig differenzierbare Funktionen approximiert, um sie numerisch besser einsetzen zu können [61, S. 666].

Um aus Messdaten eine glatte Referenzlinie zu generieren, wird ein sekundäres Optimierungsproblem gelöst, das die Abweichung zu den Messdaten im Kompromiss zur Ableitung der Krümmung minimiert [61, S. 658]. Da die Referenzlinie der Mittellinie entsprechen muss, sollte die Abweichung zu den Messdaten so gering wie möglich gehalten werden.

Für die Optimierung des Catalunya-Circuits in Barcelona werden auf einem Desktop-PC lediglich 15 Minuten bei einer Diskretisierung von 2 m benötigt. Das Modell ist zwar nicht so detailliert wie das in [20], aber es bildet einen sinnvollen Kompromiss zwischen Detaillierungsgrad und Komplexität und erlaubt weitaus genauere Berechnungen als die QSS-Methoden. [61] beschreibt mutmaßlich die erste Methode, die für eine modellbasierte Optimierung der Rundenzeit verhältnismäßig kurze Rechenzeiten erreicht.

GUTJAHR UND WERLING [34] (2015) beschreiben einen referenzlinienbasierten Ansatz, der nicht für zeitoptimale Trajektorien bestimmt ist, sondern nur der Fzg-Querführung mittels MPC dient. Er ist interessant, da er für sein kinematisches Fzg-Modell eine Approximation durchführt, auf die in Abschnitt 3.1 noch eingegangen wird. Das Modell wird im Übrigen linearisiert, um ein zeitvariantes quadratisches Optimierungsproblem (QP) formulieren zu können, das den Abstand zum Rand sowie die Krümmung der Fahrlinie innerhalb eines Zeithorizonts minimiert. Bei einem Horizont von 4 s läuft die MPC mit einer Zykluszeit von 20 ms auf einer dSpace AutoBox.

KAPANIA [48] (2016) wurde bereits in Abschnitt 2.2 erwähnt. Er führt iterativ eine separate Quer- und Längsplanung durch, indem er für eine berechnete Linie das Geschwindigkeitsprofil berechnet, und daraus wiederum eine neue Linie generiert, bei welcher Gier- und Lenkrate minimal sind. Dazu dient ein Modell mit 3 DoF, das um die Trajektorie herum linearisiert ist. Für den 4,6 km langen *Thunderhill Raceway* (CA) konvergiert das Problem nach 4 bis 5 Iterationen [48, S. 14]. Jede Iteration benötigt 30 s. Die Methode wurde in einem automatisierten Audi TTS erprobt. In [48] wird zudem ein Algorithmus vorgestellt, um den Reibwert der Strecke zu „lernen“ und so die Trajektorie von Runde zu Runde zu verbessern.

LIMA ET AL. [50] (2018) transformieren das Optimierungsproblem mittels einer Referenzlinie in den Wegbereich und maximieren wie [49, 81] die gefahrene Strecke projiziert auf die Referenzli-

nie. Dabei wird das Gütemaß in eine TAYLOR-Reihe 2. Ordnung entwickelt und das nichtlineare Einspurmodell mit linearer Reifenkennlinie um eine Referenztrajektorie herum linearisiert und zudem der Kammsche Kreis durch ein innenliegendes Polygon approximiert, um wie in [81] ein quadratisches Optimierungsproblem (QP) zu erzeugen. Um den linearen Bereich der Reifenkennlinie nicht zu verlassen, wird die Geschwindigkeit in jedem Punkt proportional zur Krümmung begrenzt, indem der Zusammenhang (2.1), also  $v_{\max} = \sqrt{|a_{y,\max}/\kappa|}$ , für das aktuelle  $\kappa$  linearisiert wird. Wie bei einem SQP-Verfahren wird das Optimierungsproblem iterativ um die aktuelle Lösung herum linearisiert. Die Online-Planung verwendet einen Planungshorizont mit  $N = 14$  Diskretisierungspunkten bei einer Schrittweite von 6 m. Für 2 Iterationen, also zweimaliges Lösen des QPs, werden in MATLAB auf einer INTEL Core i7 Mobile-CPU durchschnittlich 25 ms benötigt [50, S. 1082]. Die Planung kann dabei auch statische Hindernisse berücksichtigen.

HEILMEIER ET AL. [37] (2019) erweitern den Ansatz von [12] für eine iterative Berechnung einer krümmungsminimalen Linie, für welche anschließend mittels Vorwärts-/Rückwärts-Integration ein Geschwindigkeitsprofil ermittelt wird [37, S. 8]. Auf [37, S. 9] vergleichen sie für den Formula E-Kurs in Berlin von 2018 die Ergebnisse der krümmungsminimalen Planung mit denen einer zeitoptimalen Planung, welche auf GUNDLACH ET AL. [96] und PERANTONI ET AL. [61] basiert. Die krümmungsminimale Bahnplanung verschiebt durch Lösen eines QPs die Stützstellen der Fahrlinie, um dadurch die quadrierte Krümmung zu minimieren. Da einzelne variable Terme im QP als konstant angenommen werden müssen, wird es iterativ gelöst. Im Gegensatz zu [12] erfolgt die Verschiebung der Punkte nicht entlang der Einheitsvektoren, sondern entlang der Normalenvektoren, wie es auch bei dem vergleichbaren Filter-Algorithmus aus [96, S. 230 f.] (s. Abschnitt 6.2) der Fall ist. Im Gegensatz zu [96] werden die Ableitungen zur Berechnung der Krümmung anhand der Spline-Koeffizienten anstelle eines Differenzenquotienten gebildet. Für den 2,3 km langen Formula E-Kurs dauert die krümmungsminimale Planung bei einer konstanten Diskretisierung von 3 m auf einer INTEL Core i7 Mobile-CPU 18 s. Für die zeitoptimale Planung werden bei einem Punktmasse-Modell 37 s und bei einem Zweispurmodell 2,5 min benötigt.

### Vergleich mit der vorliegenden Arbeit

Die Methode der vorliegenden Arbeit ist generell für die zeitoptimale Trajektorienplanung eines gesamten Rundkurses ausgelegt, da eine abschnittsweise Moving-Horizon-Planung (im Sinne der Zeitoptimalität für den gesamten Rundkurs) zwangsläufig suboptimal ist, wie sich auch in [32, 50] gezeigt hat. Zusätzlich kann die Methode aber auch auf Abschnitte angewendet werden, damit die Planung, die auch statischen Hindernissen ausweichen kann, echtzeitfähig ist. Dadurch entsteht eine Moving-Horizon-Strategie, die aber im Gegensatz zu [32] keine klassische MPC darstellt, weil Regler und Planung weiterhin getrennt sind (siehe Bild 1.2). Einer der wichtigsten Aspekte dieser Arbeit ist die Reduktion der Rechenzeit, ohne wie in [31, 38, 56, 14, 12, 81, 48, 50, 37] auf QSS-Ansätze, Linearisierungen oder andere starke Modellvereinfachungen (z. B. Punktmasse) zurückzugreifen. Natürlich ist dabei eine Modellkomplexität wie in [20] mit 7 DoF unmöglich, wie auch [61] gezeigt hat. Eine detaillierte Modellierung ist auch nur dann sinnvoll, wenn alle modellierten Details, d. h. alle Parameter, hinreichend bekannt sind. Anderenfalls passiert es wie in [20], dass man zwar viele Details modelliert, aber die resultierende Linie trotzdem von der real fahrbaren

Linie abweicht, da die Parameter im Labor gemessen wurden und von den tatsächlichen abweichen. Daher muss man sich gut überlegen, welche Aspekte im Modell einen Mehrwert bringen, und welche das Modell verkomplizieren, jedoch einen ungewissen Nutzen bringen.

Hier wird ein nichtlineares Einspurmodell mit PACEJKA-Reifenkennlinie und 3 DoF verwendet, welches zusätzlich eine quasi-statische Modellierung der Achslastverteilung (mit Schwerpunkthöhe  $h_{SP} > 0$ ) und eine daraus resultierende dynamische Bremskraftverteilung aufweist. Die Antriebskraft wird statisch auf Vorder- und/oder Hinterachse verteilt. Das Reifenmodell nach PACEJKA kommt auch im Regler zum Einsatz, da es im relevanten Bereich invertierbar ist, was für den Entwurf vorteilhaft ist.

Die Rechenzeit liegt auf einer INTEL Core i7 Mobile-CPU der 4. Generation (2014) für das 4,5 km lange *Autódromo do Algarve* (PT) bei einer Diskretisierung von 1 – 5 m (1510 DP) in C++ unter 2 s und in MATLAB bei ca. 3 s. Dabei kommt die Methode ohne Startlösung aus, d. h. dass anfangs alle Zustandsvariablen und Stellgrößen null sind. Eine längere Strecke (wie die *Nordschleife*) von 20 km benötigt ca. 20 s in MATLAB. Für die Berechnung des 2,3 km langen Formula E-Kurses von 2018 in Berlin werden für einen Vergleich mit [37] die Fahrzeugparameter bestmöglich an den Formula E-Rennwagen angepasst und Maximalbeschleunigung und -geschwindigkeit auf die in [37] angegebenen Werte begrenzt. Dann dauert die Berechnung in MATLAB 1,7 s und die simulierte Rundenzeit  $T_{lap} = 85,4$  s ist  $\Delta T_{lap} = 0,5$  s langsamer als diejenige Rundenzeit, welche in [37, S. 9] für die *zeitoptimale Planung mit einem Zweispurmodell* angegeben ist, und  $\Delta T_{lap} = 0,7$  s schneller als die für die krümmungsminimale Linie angegebene Zeit.

Die Online-Planung benötigt für einen Horizont von  $N = 95$  DP (vgl. Bild 7.2 auf S. 132) durchschnittlich ca. 40 ms. Somit ist dieser Algorithmus deutlich schneller als alle zuvor angeführten, selbst wenn man eine Performance-Steigerung der Computer in den vergangenen Jahren bedenkt.

Im Gegensatz zu den meisten zitierten Quellen ist das Fzg-Modell auf seriennahe Fahrzeuge anstelle von Formel-Rennwagen ausgerichtet, was Einfluss auf die Modellierung hat. Auch ist die häufig erwähnte Optimierung von Fzg-Parametern für diese Arbeit nicht relevant. Die Methode verwendet zwar zum einen ein zeitoptimales Gütemaß, zum anderen werden aber auch Komfort-Aspekte betrachtet. Im Rahmen einer geringen Verschlechterung der Rundenzeit von weit weniger als 1 % wird die Trajektorie durch entsprechende Terme im Gütemaß möglichst glatt gewählt. Gerade diese Aspekte zeigen in Fahrversuchen eine deutliche Wirkung. Mitunter fühlen sich „Knicke“ in der Ableitung der Krümmung schon sehr unruhig an und können durch eine minimale Veränderung der Trajektorie mit marginalen Zeiteinbußen geglättet werden. Dies ist für die Online-Planung besonders wichtig, weil sich dabei die resultierende Trajektorie durch die Aneinanderreihung einzelner Abschnitte ergibt. Die Trajektorienplanung wurde mit den in Abschnitt 1.2.1 beschriebenen Fahrzeugen bei vollautomatisierter Fahrt auf diversen Strecken, u. a. auf dem *Autódromo do Algarve* (PT), dem *Autodrom Most* (CZ) und Handlings-Kursen, erprobt.

Wie [20, 61, 50] basiert die Methode auch auf einer Referenzlinie, jedoch muss diese im Gegensatz zu [20, 61] weder in der Fahrbahnmitte verlaufen noch eine konstante Diskretisierung aufweisen. Die Diskretisierung wird u. a. in Abhängigkeit der Krümmung üblicherweise im Bereich von 1 m bis 5 m gewählt. Sie ist wie in [61] für Stellgrößen und Zustände identisch.

## 3 Modellbildung und Zeitoptimalität

In Kapitel 2 wurden verschiedene Ansätze für eine Trajektorienplanung angeführt und anhand ihrer prinzipiellen Vor- und Nachteile ein präferiertes Konzept für diese Arbeit abgeleitet. Es sieht vor, den Lösungsraum der Trajektorienplanung nicht auf vorgegebene Muster einzuschränken, eine simultane Quer- und Längsplanung durch Lösen eines gemeinsamen Optimierungsproblems durchzuführen, ohne Verwendung von Ersatzgrößen die reine Rundenzeit zu minimieren und die Trajektorie über den Weg zu parametrieren.

Dieses Kapitel beschreibt zuerst alle Grundlagen der Trajektorienplanung, angefangen von dem zeitoptimalen Gütemaß und der Transformation vom Zeit- in den Wegbereich über die Kinematik des Fahrzeugs bis hin zur Modellierung der Fahrzeugdynamik. Darauf aufbauend wird in Kapitel 5 das Optimierungsproblem formuliert, nachdem zuvor in Kapitel 4 die mathematischen Grundlagen zum Lösen eines Optimierungsproblem behandelt wurden.

### 3.1 Minimierung der Rundenzeit mittels Weg-Zeit-Transformation

Das primäre Ziel der Trajektorienplanung ist die Zeitoptimalität, welche unmittelbar in einem Gütemaß abgebildet werden soll. Dies führt zu dem Optimierungsproblem

$$\min T_{\text{end}} = \int_{t=0}^{T_{\text{end}}} dt, \quad (3.1)$$

welches schwierig zu lösen ist, weil es nicht nur eine freie Endzeit sondern auch einen freien Endzustand hat. Denn der Endzustand ergibt sich erst durch die Optimierung. Transformiert man es in den Ortsbereich, so lautet es

$$\min T_{\text{end}} = \int_{s=0}^{S_{\text{end}}} \sigma(s) ds, \quad \sigma := \frac{dt}{ds}, \quad (3.2)$$

wobei  $S_{\text{end}}$  die bekannte Länge der Referenzlinie ist (vgl. Definition der Referenzlinie auf S. 7). Diese Darstellung hat gegenüber [49, 32, 81] den Vorteil, dass die zeitliche Länge des Optimierungshorizonts nicht im Vorfeld so festgelegt werden muss, dass sie maximal so lang wie die zu berechnende Fahrzeit ist. Die Transformation  $\sigma$ , die auch in [20, 61] zum Einsatz kommt, ergibt sich aus der Kinematik und wird im Folgenden anhand Bild 3.1 hergeleitet. Sie führt zudem direkt zu einer passenden mathematischen Beschreibung der Referenzlinie, welche das Bezugssystem bildet und vergleichbar mit FRENET-Koordinaten ist (Abschnitt 3.1.2).

Bild 3.1(a) zeigt ein Fahrzeug (Fzg), dessen Schwerpunkt auf der Referenzlinie liegt. Die Fzg-Längsachse  $x$  ist um den Gierwinkel  $\psi = \gamma + \psi_t$  zum ortsfesten Koordinatensystems  $(x_0, y_0)$

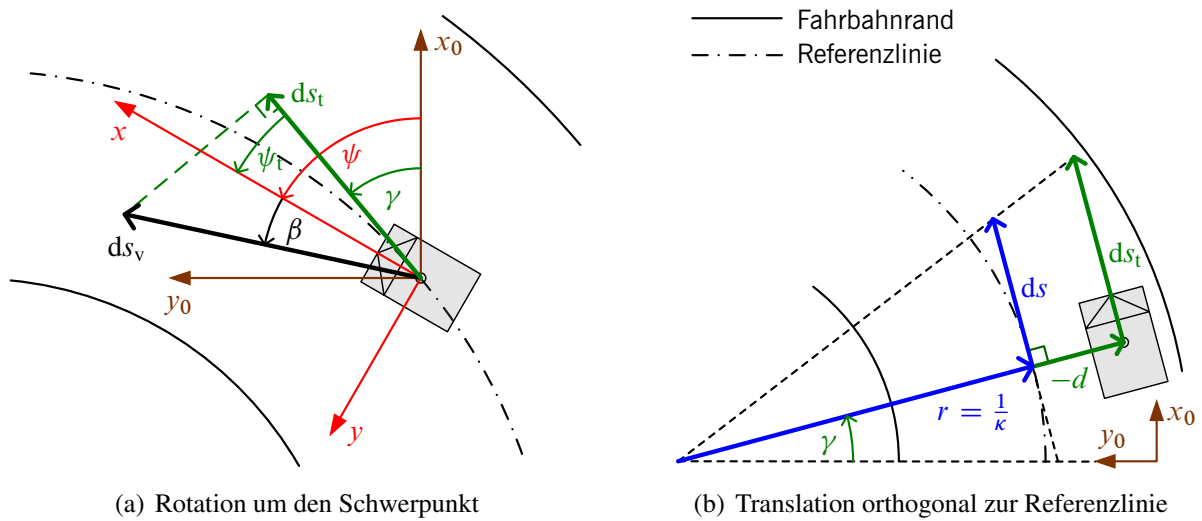


Bild 3.1: Kinematik des Fahrzeugs [95]

gedreht, welches im Folgenden *globales Koordinatensystem* genannt wird.  $\gamma$  ist die globale Orientierung der Tangente an der Referenzlinie und  $\psi_t$  der relative Gierwinkel zwischen dieser Tangente und der  $x$ -Achse. Der Schwerpunkt (SP) bewegt sich mit der Geschwindigkeit  $v$  in Richtung des Kurswinkels  $\chi = \psi + \beta$ , sodass zwischen der Fzg-Längsachse und der Bewegungsrichtung des SPs der Schwimmwinkel  $\beta$  liegt. Somit verschiebt sich der SP während eines Zeitinkrements  $dt$  um das Weginkrement

$$ds_v = v dt. \quad (3.3)$$

Um die Orientierung und Bewegungsrichtung des Fzgs zu berücksichtigen, wird  $ds_v$  auf die Referenzlinie projiziert, woraus sich

$$ds_t = \cos(\psi_t + \beta) ds_v \quad (3.4)$$

ergibt.

Bild 3.1(b) zeigt eine Translation des Fzgs zur Referenz. Der SP bewegt sich um das Weginkrement  $ds_t$  parallel zur Referenz, ist jedoch um den vorzeichenbehafteten Querversatz  $d$  orthogonal zur Referenz verschoben. Projiziert man  $ds_t$  wiederum auf die Referenz, ergibt sich durch den Strahlensatz

$$ds = \frac{r}{r - d} ds_t = \frac{1}{1 - \kappa d} ds_t, \quad (3.5)$$

worin  $\kappa = r^{-1}$  die Krümmung der Referenzlinie ist.

Zusammenfassend ergibt sich aus (3.3), (3.4) und (3.5) die Transformationsvorschrift

$$\sigma = \frac{dt}{ds} = \frac{dt}{ds_v} \frac{ds_v}{ds_t} \frac{ds_t}{ds} = \frac{1 - \kappa d}{v \cos(\psi_t + \beta)}. \quad (3.6)$$

Unter der Voraussetzung eines kontinuierlichen positiven Fortschritts auf der Referenz  $ds > 0$  und einer positiven, endlichen Fahrzeit  $dt \in (0, \infty)$  muss auch  $\sigma \in (0, \infty)$  sein. Daraus ergeben sich folgende Einschränkungen der Trajektorienplanung:

- $v > 0$  ist die größte Einschränkung, da keine Verzögerung bis in den Stillstand berechnet werden kann. Jedoch ergibt sich diese Bedingung bereits aus vielen Fahrdynamikmodellen wie dem Einspurmodell.
- $|\psi_t + \beta| < 90^\circ$  bedeutet, dass sich das Fzg im spitzen Winkel zur Referenz bewegen muss. Dies kann i. d. R. gut gewährleistet werden, indem die Referenz dem Fahrbahnverlauf folgt. (Driften wird in dieser Arbeit ebenso ausgeschlossen wie der Fall  $v < 0$  und  $|\psi_t + \beta| > 90^\circ$ , in welchem das Fzg die Strecke rückwärts abfährt.)
- $\kappa d < 1 \Leftrightarrow \text{sgn}(\kappa) d < |r|$  kann man sich mit Bild 3.1(b) veranschaulichen: In Linkskurven ist  $\kappa > 0$ . Wenn das Fzg nun in einer Linkskurve um  $d = r > 0$  links versetzt zur Referenz stünde, wäre es gerade im Kurvenmittelpunkt, und ein Fortschritt  $ds > 0$  korrespondierte mit einer Rotation um den Schwerpunkt ohne eine Translation, sodass (3.5) nicht definiert wäre. Für  $d > r$  müsste es sich sogar rückwärts bewegen, um ein  $ds > 0$  zu erzielen. Diese Bedingung muss beim Generieren der Referenz beachtet werden, indem die Referenz am besten so gewählt wird, dass ihre Krümmung stets so klein ist, dass  $|r|$  größer als der Abstand zur Kurveninnenseite ist. In der Regel ist diese Bedingung für eine (glatte) Linie in der Nähe der Fahrbahnmitte erfüllt.

Die Transformation (3.6) ist für die Verwendung im Gütemaß (3.2) plausibel und nachvollziehbar, da sie einige Grundregeln ausdrückt, um die Fahrzeit zu minimieren:

1. Schnell fahren,
2. in Kurven ( $|\kappa| > 0$ ) auf der Innenseite fahren, indem  $\text{sgn}(d) = \text{sgn}(\kappa) \Rightarrow \kappa d > 0$ , um
3. bei gleichem Fortschritt  $ds$  die Fahrstrecke  $ds_t$  zu minimieren ( $\kappa d \rightarrow 1 \xrightarrow{(3.5)} ds_t \rightarrow 0$ ), und
4. dem Streckenverlauf folgen, indem  $\cos(\psi_t + \beta) \approx 1$ .

Für Geraden ( $\kappa = 0$ ) ist der Querversatz  $d$  irrelevant. Außer der Geschwindigkeit ist dann nur ein Kurswinkel parallel zur Referenz, also  $\psi_t + \beta \approx 0$  entscheidend.

Wie in Abschnitt 2.4 angekündigt, wird noch eine Approximation aus [34] betrachtet. [34] geht für niedrige Geschwindigkeiten von einem kleinen  $d$  und für hohe von einem kleinen  $\kappa$  aus, sodass in beiden Fällen  $\kappa d \approx 0$  ist. Zudem wird eine kleine Winkeldifferenz  $\psi_t$  und ein Schwimmwinkel  $\beta \equiv 0$  angenommen, sodass  $ds \approx ds_v$  ist. Wenn man diese Approximation auf die Transformation (3.6) anwendet, ergibt sich die naheliegende Gleichung  $ds \approx v dt$ , die durchaus auch eine Approximation für  $\sigma$  darstellen könnte. Für wenig dynamische Manöver reicht eine solche Approximation aus, für Zeitoptimalität würden aber wesentliche Einflussgrößen fehlen.

### 3.1.1 Kinematik des Fahrzeugs

Die Kinematik wird durch die DGLs für die laterale Translation sowie die Rotation beschrieben. Die longitudinale Translation ergibt sich beim vorliegenden Ansatz implizit durch den unabhängigen Parameter  $s$  und bedarf keiner DGL. Die Änderung der Fzg-Position in lateraler Richtung

ist der Anteil des Geschwindigkeitsvektors senkrecht zur Referenz:

$$\dot{d} = \frac{ds_v}{dt} \sin(\psi_t + \beta) = v \sin(\psi_t + \beta) \quad (3.7)$$

Die Änderung der Orientierung ist die Ableitung des relativen Gierwinkels  $\psi_t = \psi - \gamma$ . Bild 3.1(b) zeigt, dass der Winkel zwischen dem Radius  $r$  und der  $y_0$ -Achse aufgrund von Dreiecksbeziehungen gerade  $\gamma$  ist. Somit ist

$$ds = r d\gamma \quad (3.8)$$

und

$$\dot{\psi}_t = \dot{\psi} - \dot{\gamma} = \dot{\psi} - \frac{d\gamma}{ds} \frac{ds}{dt} = \dot{\psi} - \frac{\kappa}{\sigma}. \quad (3.9)$$

Der Querversatz  $d$  und der relative Gierwinkel  $\psi_t$  spannen mit weiteren fahrdynamischen Größen (Abschnitt 3.2) den Zustandsraum für das nichtlineare Fahrzeugmodell  $\dot{\mathbf{x}} = \mathbf{f}_{\text{mod}}(\mathbf{x}, \mathbf{u})$  mit den Zuständen  $\mathbf{x}$  und den Eingängen  $\mathbf{u}$  auf. Durch eine Multiplikation mit  $\sigma$  können die zeitabhängigen DGLs des Zustandsraumsystems in den Ortsbereich transformiert werden:

$$\dot{\mathbf{x}} = \frac{d\mathbf{x}}{dt} = \mathbf{f}_{\text{mod}}(\mathbf{x}, \mathbf{u}) \quad \Rightarrow \quad \mathbf{x}' := \frac{d\mathbf{x}}{ds} = \sigma(\mathbf{x}) \mathbf{f}_{\text{mod}}(\mathbf{x}, \mathbf{u}) \quad (3.10)$$

### 3.1.2 Die Referenzlinie als Bezugssystem der Fahrlinie

Im Optimierungsproblem sollen die Fahrzeugposition und -ausrichtung sowie die Referenzlinie ausschließlich in lokalen Koordinaten beschrieben werden. Schließlich ist es für die Optimierung unerheblich, wo sich der Streckenabschnitt global befindet und wie er in Nord-/Süd-Richtung orientiert ist. Lokale Koordinaten bleiben in einer einheitlichen Größenordnung, wodurch sich manche numerischen Probleme vermeiden lassen. Dafür bietet es sich an, die Referenzlinie durch ihre Krümmung  $\kappa(s)$  und die Fahrbahnbreite durch den orthogonalen Abstand  $d_{\text{max}}(s)$  nach links und rechts zu beschreiben. Meist liegt die Referenz in der Nähe der Fahrbahnmitte, damit die numerischen Eigenschaften näherungsweise symmetrisch sind. Sie könnte aber auch neben der Fahrbahn verlaufen. Ausgangspunkt für ihre Berechnung, die in Abschnitt 6.1 dargelegt ist, sind die Fahrbahn­ränder, die als kartesische Punkte im globalen  $(x_0, y_0)$ -Koordinatensystem vorliegen.

Die Beschreibung der Fzg-Position und -ausrichtung erfolgt im Optimierungsproblem durch den Querversatz  $d(s)$  und den relativen Gierwinkel  $\psi_t(s)$  und ist somit ausschließlich auf die Referenz bezogen. Die Position in longitudinaler Richtung ergibt sich implizit durch den unabhängigen Parameter  $s$ . Um die durch  $d(s)$  und  $\psi_t(s)$  gegebene Rennlinie (engl. raceline, RL) nach der Optimierung in das globale Koordinatensystem rücktransformieren zu können, benötigt man zusätzlich die globalen Koordinaten der Referenz:

$$\begin{pmatrix} x_{\text{rl}}(s) \\ y_{\text{rl}}(s) \end{pmatrix} = \begin{pmatrix} x_{\text{ref}}(s) \\ y_{\text{ref}}(s) \end{pmatrix} + d(s) \vec{n}(s), \quad d_{\text{max, re}}(s) \leq d(s) \leq d_{\text{max, li}}(s) \quad (3.11)$$



Darin ist  $\vec{n}$  der Normalenvektor zur Referenzlinie, der sich sowohl aus ihrer Orientierung als auch aus ihrer Ableitung [53, S. 503] ergibt:

$$\vec{n}(s) = \begin{pmatrix} -\sin(\gamma(s)) \\ \cos(\gamma(s)) \end{pmatrix} = \frac{d}{ds} \begin{pmatrix} -y_{\text{ref}}(s) \\ x_{\text{ref}}(s) \end{pmatrix} \quad (3.12)$$

Aus der ersten Darstellung lässt sich  $|\vec{n}| = 1$  direkt erkennen, für die zweite ergibt sich dies aus der Differentiation nach  $ds = \sqrt{dx_{\text{ref}}^2 + dy_{\text{ref}}^2}$  über

$$|\vec{n}| = \sqrt{\left(\frac{dx_{\text{ref}}}{ds}\right)^2 + \left(\frac{dy_{\text{ref}}}{ds}\right)^2} = \sqrt{\frac{dx_{\text{ref}}^2 + dy_{\text{ref}}^2}{\sqrt{dx_{\text{ref}}^2 + dy_{\text{ref}}^2}^2}} = 1. \quad (3.13)$$

Hierfür ist es wichtig, dass  $x_{\text{ref}}(s)$  und  $y_{\text{ref}}(s)$  tatsächlich über die Weglänge  $s$  parametrisiert sind, wie es angegeben ist. Anderenfalls würde die Ableitung des Parameters nach  $s$  in (3.12) einfließen.

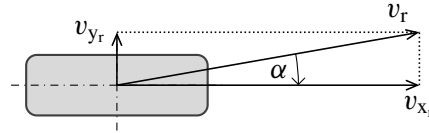
## 3.2 Modellierung der Fahrzeugdynamik

Das Fahrverhalten eines Kraftfahrzeugs, welches durch das Modell mathematisch beschrieben wird, ist die Reaktion des Fahrzeugs auf Lenken sowie auf Beschleunigen und Bremsen während der Kurvenfahrt [55, Kap. 19]. Zur Bewegungserzeugung müssen Kräfte zwischen der Fahrbahn und den Reifen durch *Schlupf* übertragen werden, welcher durch eine Scherverformung der Profilelemente entsteht [6, S. 21 ff.]. Deswegen bestimmen die Eigenschaften des Reifens das Fahrverhalten maßgeblich und bilden einen essentiellen Bestandteil des gesamten Modells. Daher werden sie zu Beginn der Modellbildung in Abschnitt 3.2.1 behandelt, bevor Abschnitt 3.2.2 das dynamische Fzg-Modell beschreibt. Anschließend werden Erweiterungen des Modells vorgenommen, die sich sowohl auf das Reifenverhalten als auch auf das restliche Modell beziehen.

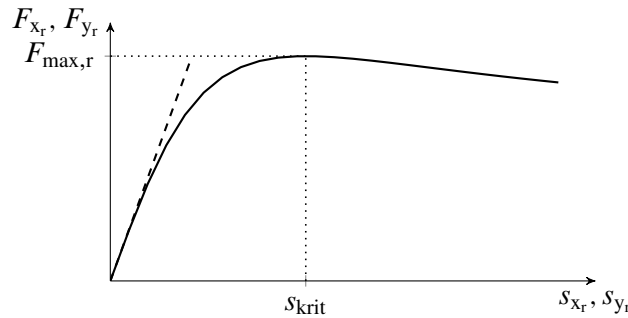
Das Fzg-Modell dominiert das gesamte Verhalten der Trajektorienplanung. Es bestimmt nämlich nicht nur die Trajektorie im Ergebnis sowie den Rechen- und Implementierungsaufwand, sondern v. a. die numerische Konvergenz der Optimierung. Diese ist für die Echtzeitfähigkeit sehr wichtig. Daher muss bei der Modellbildung ein sehr guter Kompromiss zwischen Detaillierungsgrad und Komplexität gefunden werden. Besonders wichtig dafür sind solche Approximationen, die die Qualität nur geringfügig verschlechtern, aber gleichzeitig für die Optimierung eine signifikante Verbesserung darstellen. Dieser Aspekt wird im Verlauf der Arbeit mehrfach aufgegriffen.

### 3.2.1 Reifenmodell

Die Reifenkräfte werden über die Reifenaufstandsfläche, den sogenannten *Latsch*, übertragen. Reifenkräfte und -geschwindigkeiten sind in einem reifenfesten, kartesischen Koordinatensystem  $(x_r, y_r, z_r)$  angegeben, wobei die  $x_r$ -Achse nach vorne in Reifenlängsrichtung und die  $z_r$ -Achse nach oben zeigen. Der Längsschlupf  $s_{x_r}$ , der für die Generierung der Längskraft notwendig ist,



**Bild 3.2:** Der Reifen bewegt sich in Richtung des Schräglaufwinkels.



**Bild 3.3:** Typischer Verlauf der Horizontalkraft bei reinem Längs- oder Querschlupf

bezeichnet die relative Differenz zwischen der Reifenumfangsgeschwindigkeit  $\omega_r r_{\text{dyn}}$  und der Reifenlängsgeschwindigkeit  $v_{x_r}$ , wobei  $r_{\text{dyn}}$  der dynamische Reifenhalbmesser ist. Der Längsschlupf definiert sich gemäß [59, S. 65] über die relative Geschwindigkeitsdifferenz:

$$s_{x_r} = \frac{\omega_r r_{\text{dyn}} - v_{x_r}}{v_{x_r}} \quad (3.14)$$

Um bei einer Kurvenfahrt Querkraften aufbringen zu können, muss ein Querschlupf vorliegen, indem die Bewegungsrichtung des Reifens  $v_r$  um den *Schräglaufwinkel*  $\alpha$  zur Reifenlängsachse  $x_r$  gedreht ist, wie Bild 3.2 zeigt. Nach [59, S. 67 f.] ergibt sich der Querschlupf  $s_{y_r}$  aus dem Verhältnis zwischen Quer- und Längsgeschwindigkeit. Somit kann er auch aus dem Schräglaufwinkel berechnet werden, welcher in Abschnitt 3.2.2 aus kinematischen Größen berechnet wird und eine wesentliche Größe der Fahrdynamik bildet:

$$s_{y_r} = -\frac{v_{y_r}}{v_{x_r}} = \tan(\alpha) \quad (3.15)$$

Bild 3.3 zeigt einen typischen Zusammenhang zwischen der Horizontalkraft (Umfangs-/Seitenkraft) und dem Schlupf. Dabei ergeben sich bei reinem Längs- oder reinem Querschlupf ähnliche Verläufe. Man kann erkennen, dass die Horizontalkraft bei geringem Schlupf näherungsweise linear zunimmt und bei einem kritischen Schlupf  $s_{\text{krit}}$  ihr Maximum erreicht. Mit weiter zunehmendem Schlupf gleitet der Reifen und die übertragbare Kraft nimmt ab. Bei einer zeitoptimalen Trajektorie muss an jedem Punkt mindestens ein Zustand auf einer physikalischen Grenze liegen. Neben der Motorleistung (Stellenergie) stellt die übertragbare Reifenkraft diese physikalische Grenze dar. Die Schwierigkeit dabei ist, dass die Reifenkennlinie in diesem Bereich stark nichtlinear ist.

Der Schräglaufwinkel, bei dem die maximale Querkraft entsteht, liegt bei Serienfahrzeugen meist unter  $10^\circ$ . In diesem Bereich kann der Tangens ohne nennenswerte Fehler linearisiert werden,

sodass anstelle des Querschlupfs der Schräglaufwinkel verwendet werden kann:

$$\tan(\alpha) = \alpha + \mathcal{O}(\alpha)^3 \stackrel{(3.15)}{\Rightarrow} s_{y_r} \approx \alpha \quad (3.16)$$

Selbst bei  $\alpha = 10^\circ$  ergibt sich durch die Linearisierung lediglich ein Fehler von 1 %, der aufgrund der geringen Steigung von  $F_{y_r}$  in diesem Bereich vernachlässigbar ist. Für die hier beschriebene Trajektorienplanung ist eine genaue Modellierung des Verhaltens bei  $|s_r| > s_{\text{krit}}$  irrelevant, weil die Planung von vornherein so ausgelegt ist, dass die Trajektorie immer im stabilen Bereich  $|s_r| \leq s_{\text{krit}}$  liegt und Gleiten der Reifen (Driften) ausgeschlossen ist. Hierfür gibt es drei Gründe: Erstens ist die Planung zwar für den Grenzbereich aber dennoch mit Blick auf den realen Straßenverkehr ausgelegt und dort ist Driften nicht vorgesehen. Zweitens ist Driften auch auf üblichen Rennstrecken nie zeitoptimal. Erst bei sehr engen Kurven und Niedrigreibwert, wie es z. B. bei Rallyes vorkommt, kann Driften einen Zeitvorteil bringen. Drittens ist die Modellierung eines kontrollierten (sicheren) Driftvorgangs aufgrund der Instabilität sowie den Nichtlinearitäten hochkomplex und mit diesen Modellen nicht möglich.

Liegt eine Kombination aus Längs- und Querschlupf vor, ergibt sich der resultierende Reifenschlupf  $s_r$  aus der  $\ell_2$ -Norm:

$$s_r = \sqrt{s_{x_r}^2 + s_{y_r}^2} \quad (3.17)$$

Somit kann ein Reifen, der bereits in eine Richtung Kraft überträgt, in die andere Richtung nicht mehr die volle Kraft übertragen. Diesen Zusammenhang stellt der *Kammsche Kreis* dar, der bereits in Bild 2.2 auf S. 15 beschrieben wurde.

Eine detaillierte Modellierung des Reifenverhaltens ist schwierig, weil es stark nichtlinear ist und viele Abhängigkeiten aufweist. Manche essentiellen Einflussgrößen wie die genaue Fahrbahnbeschaffenheit sind i. d. R. unbekannt. Andere Zusammenhänge liegen – wenn überhaupt – nur in Form von einmal gemessenen oder „typischen“ Kennlinien vor, wie z. B. der Einfluss von Temperatur, Feuchtigkeit, Reifendruck und Abnutzung auf die ohnehin schon komplizierten Zusammenhänge der kinetischen Größen. Viele Reifenmodelle betrachten in ihrer Grundform nur den Zusammenhang zwischen Schlupf und Horizontalkraft für einen Betriebspunkt und können gegebenenfalls erweitert werden, indem man ihre Parametrierung abhängig von weiteren Einflussgrößen wählt. Eine Übersicht über verschiedene Modelle befindet sich in [5, S. 25 ff.] und [45, Kap. 2.6]. Die Modellierung ist – wie immer – ein Kompromiss zwischen Rechenaufwand und Genauigkeit, wobei es keinen Sinn hat, Zusammenhänge im Detail modellieren zu wollen, für die man keine zuverlässigen Parameter des realen Systems kennt.

Im einfachsten Fall wählt man ein linearisiertes Reifenmodell (gestrichelte Linie in Bild 3.3), das eine Approximation bei geringem Schlupf bietet, aber im Grenzbereich viel zu optimistisch ist. Neben dem Modell nach BURCKHARDT [18, Kap. 1.4], auf das hier nicht eingegangen wird, ist die *Magic Formula* nach PACEJKA [59] das am weitesten verbreitete Modell. Darüber hinaus wird hier noch ein gebrochen rationaler Ansatz aus [5, S. 25 ff.] vorgestellt.

### Magic-Formula-Reifenmodell

Die *Magic Formula* nach PACEJKA [59] basiert nicht auf physikalischen Betrachtungen, sondern wurde empirisch entwickelt. Sie beschreibt alle wesentlichen stationären Aspekte des Reifens über einen numerischen Zusammenhang in Form eines geschlossenen mathematischen Ausdrucks, der beliebig oft stetig differenzierbar ist. Die Magic Formula

$$F_r(x) = D \sin \left( C \arctan \left( Bx - E \left( Bx - \arctan(Bx) \right) \right) \right) \quad (3.18)$$

bildet einen Eingang  $x$ , welcher entweder  $x \equiv s_{x_r}$  oder  $x \equiv s_{y_r} \approx \alpha$  sein kann, auf  $F_{x_r}$  bzw.  $F_{y_r}$  (oder auf das Rückstellmoment) ab [59, Kap. 4.3]. Darin sind  $B$ ,  $C$  und  $E$  Formfaktoren („Reifenparameter“). Das Maximum der Reifenkraft liegt bei

$$F_{\max,r} = D = \mu F_{z_r}. \quad (3.19)$$

Das Ergebnis entspricht dem Verlauf in Bild 3.3. Zusätzlich kann die Kurve horizontal oder vertikal um einen Offset verschoben werden, um Richtungsabhängigkeiten des Reifens zu berücksichtigen, was in der vorliegenden Arbeit nicht geschieht. Die Anfangssteigung der Kennlinie (gestrichelte Linie in Bild 3.3) ist die Reifensteifigkeit  $c_F$ :

$$c_F = \left. \frac{dF_r}{ds_r} \right|_{s_r=0} = BCD \quad (3.20)$$

Mit dem Formfaktor  $E$  kann man das Maximum entlang der Abszisse verschieben, indem man

$$E = \frac{Bs_{\text{krit}} - \tan\left(\frac{\pi}{2C}\right)}{Bs_{\text{krit}} - \arctan(Bs_{\text{krit}})} \quad (3.21)$$

setzt [59, S. 166]. Der Parameter  $E$  hat für  $|s_r| \ll s_{\text{krit}}$  wenig Einfluss, wie man erkennt, wenn man den Arkustangens in der inneren Klammer von (3.18) in eine Taylorreihe 1. Ordnung entwickelt:

$$\arctan(Bx) = Bx + \mathcal{O}(x)^3 \quad (3.22)$$

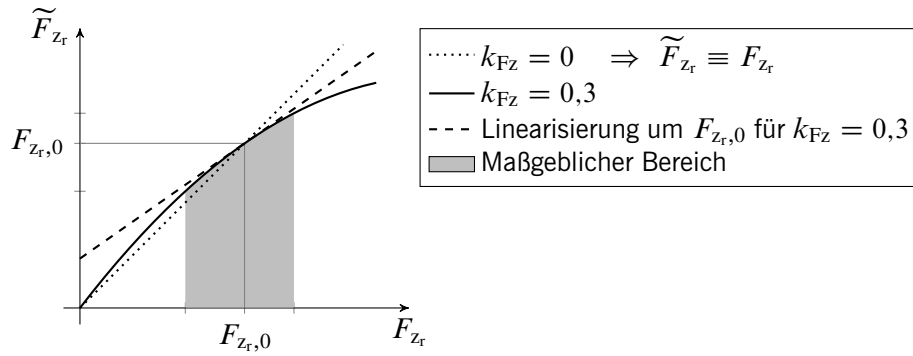
Dadurch hebt sich die innere Klammer in (3.18) weg, sodass die Horizontalkraft über

$$F_r(x) \approx D \sin \left( C \arctan(Bx) \right) \quad (3.23)$$

approximiert werden kann, was strukturell eine erhebliche Vereinfachung zu (3.18) darstellt, ohne den Verlauf im relevanten Bereich übermäßig zu beeinflussen. Das Maximum liegt dann bei

$$s_{\text{krit}} = \frac{1}{B} \tan \left( \frac{\pi}{2C} \right), \quad (3.24)$$

was in (3.21) eingesetzt zwangsläufig zu  $E = 0$  führen würde. An (3.24) erkennt man, dass  $C > 1$  gewählt werden muss, damit die Kennlinie ein Maximum besitzt und nicht monoton steigt.



**Bild 3.4:** Der Einfluss der Achslast  $F_{zr}$  auf die Horizontalkraft ist degressiv und kann gemäß (3.27) mit dem Parameter  $k_{Fz}$  modelliert werden. Meist liegt  $F_{zr}$  im grau hinterlegten Bereich.

### Gebrochen rationales Reifenmodell

Ein alternativer Ansatz, der durch seine numerisch einfach auszuwertenden Funktionen und die geringe Anzahl an Parametern besticht, ist ein Ansatz aus [5, S. 25 ff.], der dort trotz des Betrags im Nenner als „gebrochen rational“ bezeichnet wird:

$$F_{yr}(\alpha) = \frac{\tilde{c}_F \mu F_{zr} \alpha}{\mu \left( \frac{|\alpha|}{\alpha_{krit}} - 1 \right)^2 + \tilde{c}_F |\alpha|}, \quad \tilde{c}_F = \frac{c_F}{F_{zr}} \quad (3.25)$$

Vor allem im Bereich  $|\alpha| \leq \alpha_{krit}$  zeigen sich bei entsprechender Parametrierung nur kleine Abweichungen zur *Magic Formula*. Im Bereich  $|\alpha| > \alpha_{krit}$  fällt der Verlauf tendenziell stärker ab. Für die Verwendung in der Trajektorienplanung mittels Optimierung hat er leider den Nachteil, dass er aufgrund der Betragsfunktion nur einmal stetig differenzierbar ist:

$$\lim_{\alpha \rightarrow 0^+} \frac{d^2}{d\alpha^2} F_{yr}(\alpha) = - \lim_{\alpha \rightarrow 0^-} \frac{d^2}{d\alpha^2} F_{yr}(\alpha) = 2\tilde{c}_F F_{zr} \left( \frac{2}{\alpha_{krit}} - \frac{\tilde{c}_F}{\mu} \right) \quad (3.26)$$

### Einfluss der Achslast

Gleichungen (3.19) und (3.25) legen nahe, dass eine Veränderung der Achslast  $F_{zr}$  lediglich zu einer Umskalierung der Ordinate in Bild 3.3 führt, wodurch sich die Reifensteifigkeit und die Maximalkraft proportional zu  $F_{zr}$  verändern, aber das Maximum an derselben Stelle  $s_{krit}$  bleibt. Diese Proportionalität, die man von COULOMBScher Reibung kennt, gilt jedoch nur (näherungsweise) bis zur statischen Betriebslast  $F_{zr} = F_{zr,0}$ . Bei weiter zunehmender Radlast nimmt  $F_{max,r}$  nur noch schwach degressiv mit  $F_{zr}$  zu [75, S. 161]. Die degressive Zunahme kann über die leicht modifizierte Wahl des Parameters  $D = \mu \tilde{F}_{zr}$  modelliert werden, indem in Anlehnung an [6, S. 28]

$$\tilde{F}_{zr} = F_{zr} \left( 1 - k_{Fz} \frac{F_{zr} - F_{zr,0}}{F_{zr,0}} \right) = (1 + k_{Fz}) F_{zr} - \frac{k_{Fz}}{F_{zr,0}} F_{zr}^2, \quad k_{Fz} \in (0, 1) \quad (3.27)$$

gesetzt wird. Der Zusammenhang ist in Bild 3.4 abgebildet.

Durch den nichtlinearen Einfluss von  $F_{z_r}$  ist die Reifenkennlinie also nicht nur vom Reifen selbst, sondern auch von der Achslast, welche durch das Fahrwerk bestimmt wird, abhängig (weitere Fahrwerkeinflüsse, wie z. B. Wankabstützung, sind in der zitierten Literatur beschrieben). So ergeben sich meistens unterschiedliche Kennlinien für Vorder- und Hinterachse. Das Reifenmodell für Vorder- und Hinterachse wird jeweils für die statische Betriebslast parametrisiert. Im dynamischen Fahrbetrieb ist die Achslast aber nicht konstant (s. Abschnitt 3.2.3). Typischerweise ist ihre Veränderung so gering, dass (3.27) um  $F_{z_r,0}$  herum linearisiert werden kann:

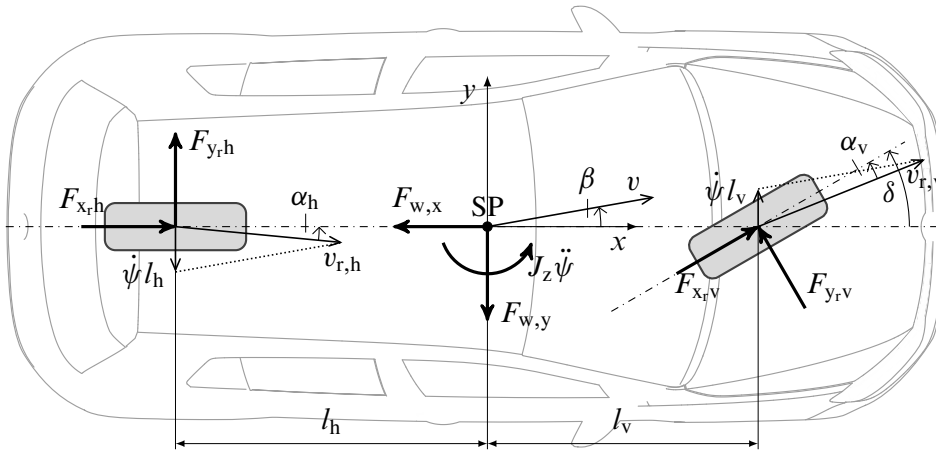
$$\tilde{F}_{z_r} \approx F_{z_r,0} + (1 - k_{Fz}) \Delta F_{z_r}, \quad \Delta F_{z_r} = F_{z_r} - F_{z_r,0} \quad (3.28)$$

Der linearisierte Zusammenhang ist in Bild 3.4 gestrichelt dargestellt. Er weicht in dem maßgeblichen Bereich kaum von der nichtlinearen Funktion ab. In (3.52) werden wir aber sehen, dass diese Linearisierung eine große Vereinfachung für das Optimierungsproblem darstellen wird.

### 3.2.2 Nichtlineares Einspurmodell

Für viele fahrdynamische Betrachtungen ist die Summe der Kräfte an einer Achse wichtiger als die einzelnen Reifenkräfte. Deswegen führt das Einspurmodell (im Gegensatz zum Zweispurmodell) die Radaufstandspunkte achsweise zusammen und rechnet mit je einem Rad in der Mitte der Achse, wie es in Bild 3.5 auf der nächsten Seite dargestellt ist. Bei dem weit verbreiteten linearen Einspurmodell (entwickelt 1940 von RIEKERT und SCHUNCK, siehe [54]) liegt der Schwerpunkt in der Fahrbahnebene, Winkelbeziehungen werden linearisiert, die Reifenkräfte sind linear vom Schräglaufwinkel abhängig und der Einfluss der Geschwindigkeit auf die Querdynamik wird quasi-statisch modelliert. Die Vereinfachungen sind jedoch nicht unbedingt notwendig. Im nichtlinearen Fall spricht man auch vom erweiterten Einspurmodell.

Die Annahme, dass der Schwerpunkt (SP) in der Fahrbahnebene liegt, hat zur Folge, dass das Fzg weder nickt (Drehung um die  $y$ -Achse) noch wankt (Drehung um die  $x$ -Achse). Nicken entsteht beim Verzögern und Beschleunigen, Wanken bei Kurvenfahrten. Weil die dabei entstehenden Drehwinkel klein sind, ist die Verschiebung des Schwerpunkts gering und kann für das Fahrverhalten oft vernachlässigt werden. Jedoch ist der Auslöser hierfür ein Nick- bzw. Wankmoment, das zu einer asymmetrischen Radlastverteilung führt. Die Achslastverteilung durch ein Nickmoment kann auch im Einspurmodell berücksichtigt werden (Abschnitt 3.2.3). Ein Wankmoment führt zu einer Belastung des kurvenäußeren und einer Entlastung des kurveninneren Rads. (Jedoch hat dabei die Fahrwerksauslegung großen Einfluss, weil der Querstabilisator einem Wankmoment entgegen wirkt.) Geht man von einem linearen Zusammenhang zwischen Aufstandskraft  $F_{z_r}$  und Querkraft  $F_{y_r}$  aus, bleibt die Summe der Querkräfte achsweise konstant. Da der Zusammenhang aber gemäß (3.27) degressiv ist, ist die tatsächliche Summe der Querkräfte kleiner als die Querkraft bei gemittelter Radlast. Um die Wankdynamik zu modellieren ohne gleich ein Zweispurmodell verwenden zu müssen, kann man den Wankwinkel als weiteren Zustand im Einspurmodell einführen und daraus einen effektiven Schräglaufwinkel berechnen, wie es in [40, S. 50 ff.] beschrieben ist. Hierauf wurde in dieser Arbeit aber aufgrund der erhöhten Modellordnung zugunsten der Rechenzeit verzichtet.



**Bild 3.5:** Einspurmodell mit dem fahrzeugfesten Koordinatensystem im SP  $(x, y)$  und den beiden radfesten Koordinaten  $(x_r, y_r)$  für Vorder- und Hinterachse. Die Pfeiltypen haben folgende Bedeutungen:

- Koordinaten und Abmessungen,
- Kräfte- und Momentenvektoren,
- Geschwindigkeitsvektoren,
- Winkel.

Trotz der teils vernachlässigten Dynamik und der approximierten Reifenkräfte wird das lineare Einspurmodell sogar bei fahrdynamisch anspruchsvolleren Systemen wie dem ESP eingesetzt [67, S. 71]. Nach [67] lässt sich das Fahrverhalten bis  $4 \text{ m/s}^2$  generell und im Übergangsbereich bis  $6 \text{ m/s}^2$  teilweise linear beschreiben. In dieser Arbeit wird allerdings ein nichtlineares Einspurmodell verwendet, das um eine approximative Achslastverteilung erweitert wird.

### Bewegungsgleichungen des nichtlinearen Einspurmodells

In Bild 3.5 sind die horizontalen Kräfte und Geschwindigkeiten eingezeichnet, aus denen nun die Bewegungsgleichungen des Fahrzeugs hergeleitet werden. Unter Berücksichtigung der Schräglaufwinkel vorne  $\alpha_v$  und hinten  $\alpha_h$  bewegen sich die Räder in Richtung der Vektoren  $v_{r,v}$  bzw.  $v_{r,h}$ . Die SP-Geschwindigkeit  $v$  ist tangential zu seiner Bahnkurve in Richtung des *Kurswinkels*

$$\chi := \psi + \beta \quad (3.29)$$

im globalen  $(x_0, y_0)$ -Koordinatensystem orientiert (vgl. Bild 3.1(a) auf S. 26). Der momentane Radius der Bahnkurve betrage  $\rho$ . Ihr Mittelpunkt ergibt sich geometrisch aus dem Schnittpunkt der Orthogonalen an  $v_{r,v}$  und  $v_{r,h}$  (nicht in Bild 3.5 eingezeichnet). Wenn sich der SP auf dieser Bahnkurve in der Zeit  $dt$  gemäß (3.3) auf S. 26 um den Weg  $ds_v$  bewegt, resultiert daraus folgende Änderung im Kurswinkel:

$$ds_v = v dt = \rho d\chi \quad \Rightarrow \quad \dot{\chi} = \dot{\psi} + \dot{\beta} = \frac{v}{\rho} \quad (3.30)$$

Mit (3.30) lässt sich die Zentripetalbeschleunigung  $a_c$ , die auf den SP in Richtung des Krümmungsmittelpunkts wirkt, als

$$a_c = \frac{v^2}{\rho} = v(\dot{\psi} + \dot{\beta}) \quad (3.31)$$

angeben. Hiermit lassen sich nun die Bewegungsgleichungen der 3 Freiheitsgrade in der Ebene für ein Fzg mit der Masse  $m$  und den Größen aus Bild 3.5 angeben:

$$0 = mv(\dot{\beta} + \dot{\psi}) \sin(\beta) - m\dot{v} \cos(\beta) + F_{x_{rv}} \cos(\delta) + F_{x_{rh}} - F_{y_{rv}} \sin(\delta) - F_{w,x} \quad (3.32a)$$

$$0 = -mv(\dot{\beta} + \dot{\psi}) \cos(\beta) - m\dot{v} \sin(\beta) + F_{x_{rv}} \sin(\delta) + F_{y_{rv}} \cos(\delta) + F_{y_{rh}} - F_{w,y} \quad (3.32b)$$

$$0 = J_z \ddot{\psi} - (F_{x_{rv}} \sin(\delta) + F_{y_{rv}} \cos(\delta)) l_v + F_{y_{rh}} l_h \quad (3.32c)$$

Darin fassen  $F_{w,x}$  und  $F_{w,y}$  die Widerstandskräfte in Fzg-Längs- bzw. -Querrichtung zusammen, die später auf S. 39 behandelt werden. Mit den drei Additionstheoremen [16, S. 80]

$$\sin(\beta)^2 + \cos(\beta)^2 = 1, \quad (3.33a)$$

$$\sin(\beta) \sin(\delta) + \cos(\beta) \cos(\delta) = \cos(\beta - \delta), \quad (3.33b)$$

$$\sin(\beta) \cos(\delta) - \cos(\beta) \sin(\delta) = \sin(\beta - \delta) \quad (3.33c)$$

werden (3.32a) und (3.32b) nach  $\dot{v}$  und  $\dot{\beta}$  aufgelöst:

$$\begin{aligned} \dot{v} = \frac{1}{m} & (F_{x_{rv}} \cos(\beta - \delta) + (F_{x_{rh}} - F_{w,x}) \cos(\beta) + \dots \\ & F_{y_{rv}} \sin(\beta - \delta) + (F_{y_{rh}} - F_{w,y}) \sin(\beta)) \end{aligned} \quad (3.34a)$$

$$\begin{aligned} \dot{\beta} = \frac{1}{mv} & (-F_{x_{rv}} \sin(\beta - \delta) - (F_{x_{rh}} - F_{w,x}) \sin(\beta) + \dots \\ & F_{y_{rv}} \cos(\beta - \delta) + (F_{y_{rh}} - F_{w,y}) \cos(\beta)) - \dot{\psi} \end{aligned} \quad (3.34b)$$

So lässt sich aus (3.7) und (3.9) sowie (3.34a), (3.34b) und (3.32c) mit dem Zustandsvektor

$$\mathbf{x} = \begin{bmatrix} v & d & \beta & \psi_t & \dot{\psi} \end{bmatrix}^\top$$

und mithilfe der Transformation (3.10) die nichtlineare Zustandsraumdarstellung

$$\begin{bmatrix} v' \\ d' \\ \beta' \\ \psi_t' \\ \dot{\psi}' \end{bmatrix} = \frac{1 - \kappa d}{v \cos(\psi_t + \beta)} \begin{bmatrix} \dot{v}(\mathbf{x}) \\ v \sin(\psi_t + \beta) \\ \dot{\beta}(\mathbf{x}) \\ \dot{\psi} - \frac{\kappa}{\sigma(\mathbf{x})} \\ \frac{1}{J_z} ((F_{x_{rv}} \sin(\delta) + F_{y_{rv}} \cos(\delta)) l_v - F_{y_{rh}} l_h) \end{bmatrix} \quad (3.35)$$

angeben, wobei die Stellgrößen noch in den Reifenkräften impliziert sind.

Im Zustandsraummodell (3.35) ist also nur die Fzg-Pose ( $d, \psi_t$ ) relativ zur Referenzlinie bzw. zum FRENET-Pfad angegeben. Die übrigen Größen, wie  $v, \dot{\psi}$  und die darin vorkommenden Kraft-/Beschleunigungsvektoren, sind absolut definiert, also nicht relativ zum FRENET-System. So entfällt die aufwendige Transformation dieser kinematischen Größen mittels der FRENETSchen Formeln [16, Kap. 3.6], was einen Vorteil gegenüber einem zeitlich parametrisierten System in FRENET-Koordinaten darstellt.



### Reifenquerkräfte

Die Reifenquerkräfte  $F_{yrv}$  und  $F_{yvh}$  in (3.35) werden jeweils nach der *Magic Formula* (3.23) mit je 3 Parametern in Abhängigkeit der Schräglaufwinkel berechnet. Die Schräglaufwinkel lassen sich aus Bild 3.5 ablesen (die gepunkteten Hilfslinien sind Parallelverschiebungen des  $v$ -Vektors):

$$\tan(\alpha_h) = \frac{\dot{\psi} l_h - v \sin(\beta)}{v \cos(\beta)} \Rightarrow \alpha_h \approx -\beta + \frac{\dot{\psi} l_h}{v} \quad (3.36a)$$

$$\tan(\delta - \alpha_v) = \frac{\dot{\psi} l_v + v \sin(\beta)}{v \cos(\beta)} \Rightarrow \alpha_v \approx \delta - \beta - \frac{\dot{\psi} l_v}{v} \quad (3.36b)$$

Die Kleinwinkelnäherung, die schon in (3.16) genutzt wird, ist auch im Grenzbereich ungefähr gültig, da Schräglauf- und Schwimmwinkel relativ klein bleiben [55, Kap. 27.1]. Da ihre Abweichung geringfügig ist und sie allgemein üblich ist, wird sie im Folgenden formal exakt gleich gesetzt (= statt  $\approx$ ).

In [55, Kap. 27.1] werden beim Momentensatz (3.32c) noch Rückstellmomente an den Reifen berücksichtigt. Der Reifenlatsch ist nicht symmetrisch zum geometrischen Radmittelpunkt, sodass der Angriffspunkt der Kraft um den sogenannten *Reifennachlauf* nach hinten versetzt ist. Dadurch entsteht ein Rückstellmoment, das versucht, den Schräglaufwinkel zu reduzieren. Es führt zu dem bekannten Phänomen, dass sich die Räder gerade ausrichten, wenn kein Moment auf das Lenkrad gegeben wird. Es unterliegt nach (3.18) einem ähnlichen Verlauf wie die Querkraft, allerdings sind seine Auswirkungen auf (3.32c) bei einem typischen Reifennachlauf  $n_r$  von wenigen Zentimetern [55, S. 40] gering, weil sein Hebelarm  $n_r \ll l_v \lesssim l_h$  ist. Das Lenkmoment ist in dieser Arbeit bei der Modellierung irrelevant, weil der Lenkwinkel  $\delta$  (= Rad-Lenkwinkel, nicht Lenkrad-Winkel) eine Eingangsgröße des Einspurmodells ist. Daher können Rückstellmomente ignoriert werden.

Das Reifenmodell nach Abschnitt 3.2.1 beschreibt das stationäre Verhalten. Da die Querkraft durch Profilverformung im Reifenlatsch erzeugt wird, welche bei einem veränderten Schräglaufwinkel erst entstehen muss, steht die Querkraft nicht sofort zur Verfügung. Der verzögerte Kraftaufbau ist wegabhängig und wird durch die *Einlauflänge*  $\sigma_\alpha$  charakterisiert [26, S. 91 f.]. Dieses transiente Verhalten wird oft durch ein PT<sub>1</sub>-Glied modelliert, dessen Zeitkonstante  $\tau = \sigma_\alpha/v$  von der Geschwindigkeit abhängt. Anders formuliert hat der Reifen nach Zurücklegen der Einlauflänge ca. 63 % der stationären Querkraft aufgebaut.

Es stellt sich die Frage, ob das transiente Verhalten bei der diskreten Trajektorienplanung berücksichtigt werden muss oder ob es statisch modelliert werden kann. Die zusätzliche Dynamik erhöht die Modellordnung um 2 (für vorne und hinten) und führt somit zu einer längeren Rechenzeit. EINSLE [23] hat das transiente Übertragungsverhalten theoretisch und messtechnisch für einige gängige Markenreifen untersucht. Demnach steigt die Einlauflänge linear mit der Radlast um ca. 0,11 m/kN, während sie pro 1° Erhöhung des Schräglaufwinkelsprungs um ca. 10 % sinkt [23, S. 67 ff.]. Alle anderen Einflüsse sind vernachlässigbar. Gehen wir von einer Radlast im Bereich von 4–6 kN aus (Näherung für Fahrzeuge wie den hier verwendeten Versuchsträgern), so ergibt sich  $\sigma_\alpha \approx 0,4 \dots 0,7$  m für kleine Schräglaufwinkel-Änderungen. Bei großen Änderungen nimmt  $\sigma_\alpha$  linear ab, bis es bei großen Sprüngen von ca. 10° nur noch bei  $\sigma_\alpha \approx 0,1 \dots 0,2$  m liegt.

Die Optimierung liefert für jeden Diskretisierungspunkt (DP) einen Schräglaufwinkel, der stationär zur gewünschten Querkraft führt. Eine transiente Modellierung ist angebracht, wenn in einem Abtastschritt der stationäre Wert nicht (näherungsweise) erreicht wird. Bei einem  $PT_1$ -Glied kann man hierfür die dreifache Zeitkonstante verwenden, sodass 95 % der stationären Querkraft vorliegen. Hier sind das  $3\sigma_\alpha = 1,2 \dots 2,1$  m, die einer Diskretisierung von 1 – 5 m gegenüberstehen. Für eine Optimalsteuerung wäre die Bedingung also nur knapp oder nicht mehr erfüllt. Es gilt jedoch zu bedenken, dass die Optimierung zwischen aufeinanderfolgenden DP nur kleine Änderungen in  $\alpha$  vornimmt (i. d. R. ist  $|\Delta\alpha| < 1^\circ$ ), was zu dem großen  $\sigma_\alpha$  führt. Aber der Trajektorienfolgeregler, der höherfrequent läuft, hat die Möglichkeit, mit einem entsprechenden (ggf. vorgesteuerten) Lenkwinkel den Schräglaufwinkel so zu beeinflussen, dass der Querkraftverlauf dem gewünschten Verlauf hinreichend genau folgt. Auf Basis dieser Überlegung wird der Reifenkraftaufbau im Planer zugunsten der Rechenzeit quasi-statisch modelliert.

### Reifenlängskräfte

In Analogie zu den Querkraften, wo der Lenkwinkel mit dem daraus resultierenden Schräglaufwinkel als Eingangsgröße dient, würde sich in Längsrichtung die Raddrehzahl mit dem daraus resultierenden Schlupf anbieten, welcher sich abhängig vom Radmoment einstellt. Doch während die Reifenkennlinie in Querrichtung wegen der wichtigen kinematischen Beziehung des Schräglaufwinkels (3.36) signifikanten Einfluss auf das Fahrverhalten hat, sind Raddrehzahl und Längsschlupf (3.14) weniger relevant, *solange* die fahrdynamischen Grenzen eingehalten werden. Deswegen wird in dieser Arbeit – wiederum zugunsten der Rechenzeit – als Eingangsgröße ein Äquivalent zur Reifenlängskraft verwendet.

Der Tatsache, dass sich die Reifenkennlinie abhängig vom kombinierten Schlupf (3.17) auf S. 31 verändert, wird durch den Kammschen Kreis Rechnung getragen. Dieses Vorgehen ist allgemein üblich. Denn eine kombinierte Schlupf-Betrachtung ist für eine Optimalsteuerung sehr komplex. Sie ist eher dann geeignet, wenn eine Richtung (z. B. die Bahn) bereits vorgegeben ist, und die andere (z. B. das Geschwindigkeitsprofil) optimiert wird (vgl. [14]). Dann können entsprechende Kennlinien in einem mehrdimensionalen GG-Diagramm hinterlegt werden.

Die Reifenlängskräfte  $F_{x,v}$  und  $F_{x,h}$  setzen sich jeweils aus den beiden Eingangsgrößen des Modells, nämlich einer Antriebskraft  $F_M \geq 0$  und einer Bremskraft  $F_B \leq 0$ , zusammen. Diese verteilen sich über eine konstante Motorkraftverteilung  $k_{MV} \in [0, 1]$  und eine dynamische Bremskraftverteilung  $k_{BV}(a_x) \in (0, 1)$  auf Vorder- und Hinterachse:

$$F_{x,v} = k_{MV} F_M + k_{BV}(a_x) F_B \quad (3.37a)$$

$$F_{x,h} = (1 - k_{MV}) F_M + (1 - k_{BV}(a_x)) F_B \quad (3.37b)$$

Zwei separate Eingangsgrößen für Antrieb und Bremse einzuführen, ist recht ungewöhnlich. Die meisten Ansätze, darunter [13, 20, 32, 61, 81], verwenden eine Variable, die für positive Werte antreibt und für negative bremst. Das hätte den Vorteil, dass die Anzahl der Optimierungsvariablen minimal gehalten wird und sich Antrieb und Bremse per se ausschließen. Um aber (3.37a) mit

$k_{MV} \neq k_{BV}$  abzubilden, müsste eine einzelne Stellgröße  $\tilde{F}$  quasi aufgesplittet werden, indem man

$$\tilde{F}_{xv} = k_{MV} \max\{\tilde{F}, 0\} + k_{BV} \min\{\tilde{F}, 0\} \quad (3.38)$$

ansetzen würde. Das hätte den Nachteil, dass  $\tilde{F}_{xv}$  wegen der Minimum-/Maximum-Funktion nicht differenzierbar wäre, worunter die Konvergenz der Optimierung leidet. Selbst wenn man sie mit einem weichen Übergang wie in [61] approximiert, hätte ihre TAYLOR-Reihe im NEWTON-Verfahren (vgl. Abschnitt 4.2.1) eine große Abweichung vom tatsächlichen Verlauf, wodurch die Liniensuche ineffizienter ablief. Deswegen werden hier separate Eingangsgrößen eingeführt.

Hierbei muss aber sicher gestellt werden, dass sich  $F_M$  und  $F_B$  ausschließen, also

$$F_M = 0 \vee F_B = 0. \quad (3.39)$$

Diese Komplementärbedingung wird im Optimierungsproblem durch die Forderung

$$(F_M F_B)^2 \cong 0 \quad (3.40)$$

abgebildet (s. Abschnitt 5.2), wobei das Quadrat lediglich der Verbesserung der Konvergenz dient. Eine etwaige kleine negative Kraft  $F_M < 0$  durch ein Motorschleppmoment ist bei der Planung von Rennlinien irrelevant, da man auf der Rennstrecke nicht im Schubbetrieb fährt.

### Antriebsleistung und Fahrwiderstände

Die Reifenlängskraft gibt ein Fahrer nur indirekt vor, indem er über die Fahrpedalstellung ein Motormoment bestimmt, das unter Berücksichtigung des Getriebes und diverser Verluste schließlich zu einem Radmoment führt. Es ist nicht notwendig, dieses ganze System in der Trajektorienplanung zu modellieren, da der Trajektorienfolgeregler mit seinen unterlagerten Regelkreisen es ermöglicht, die Systemgrenze der Planung in Richtung Reifenkräfte zu verschieben. Hierfür muss allerdings gewährleistet sein, dass der Regelkreis physikalisch in der Lage ist, dem geforderten Verlauf zu folgen. Das erfordert eine entsprechende Abstimmung der Planung, auf die später eingegangen wird. Unter der Annahme eines *idealen Lieferkennfelds* ist die Antriebskraft  $F_M$  aus einer Antriebsquelle mit konstanter Leistung  $P_M$  eine Hyperbel [15, Kap. 3.2]:

$$F_M = \frac{P_M}{v} \quad (3.41)$$

Im Stillstand ist die verfügbare Kraft nach (3.41) unendlich hoch und wird nur durch das Moment begrenzt, das der Motor maximal liefern kann. Da  $P_M$  nicht die vom Motor umgesetzte Leistung, sondern die hinter dem Getriebe abrufbare Leistung ist, sind Leistungsverluste im Antriebsstrang und Verbraucher zahlenmäßig berücksichtigt. Das ideale Lieferkennfeld geht von einem ideal schaltenden Getriebe aus, indem  $P_M = \text{const}$  angenommen wird. Jedoch steht nicht die gesamte Antriebskraft  $F_M$  zur Fzg-Beschleunigung zur Verfügung, da Widerstandskräfte überwunden werden müssen. Sie bestimmen – dominiert durch den Luftwiderstand – die Höchstgeschwindigkeit.

Zu den modellierten Fahrwiderständen  $F_{w,x}$  und  $F_{w,y}$  gehören der Luftwiderstand  $F_{Lx}$  in Längsrichtung sowie die Steigungswiderstände für die Fahrbahnneigungswinkel in Längsrichtung  $\theta$  und

in Querrichtung  $\phi$ . Die Orientierung des Fzgs zur Neigung, die mit der Winkeldifferenz  $\Delta\psi_t$  bezeichnet wird, hängt davon ab, für welche Ausgangslinie  $\theta$  und  $\phi$  vorliegen. Liegen sie für die Referenz vor, ist  $\Delta\psi_t = \psi_t$ . Der Rollwiderstand der Reifen wird vernachlässigt, weil er auf Asphalt typischerweise nur 1 % der Normalkraft [15, S. 46 ff.] beträgt. Die Massenträgheit der Räder, die einer Beschleunigung entgegenwirkt, wird in der Planung ebenfalls vernachlässigt. Es folgt:

$$F_{w,x} = F_{Lx} + mg \sin(\theta) \cos(\Delta\psi_t) + mg \sin(\phi) \sin(\Delta\psi_t), \quad (3.42a)$$

$$F_{w,y} = mg \sin(\phi) \cos(\Delta\psi_t) - mg \sin(\theta) \sin(\Delta\psi_t) \quad (3.42b)$$

Der Luftwiderstand nimmt hauptsächlich quadratisch mit der Differenz von Fzg- und Anströmgeschwindigkeit zu. Sein kubischer Anteil ist so gering, dass er im Geschwindigkeitsbereich üblicher Serienfahrzeuge problemlos vernachlässigt werden kann. Da Anströmgeschwindigkeit und Anströmwinkel unbekannt sind, werden beide zu null angenommen, sodass der Luftwiderstand quadratisch mit der Fzg-Geschwindigkeit und abhängig vom Luftwiderstandsbeiwert  $c_w$  zunimmt. Mit der Querspanfläche des Fzgs  $A$  und dem Luftdruck  $\rho_L = \text{const}$  ist [55, Kap. 3]

$$F_{Lx} = \underbrace{\frac{1}{2} c_w A \rho_L}_{=: k_{Lx}} v^2. \quad (3.43)$$

Die Fahrbahnneigungswinkel (engl. grade/banking), die sich aufgrund des Höhenprofils der Strecke ergeben, erzeugen aus der Gewichtskraft  $mg$  eine Komponente in  $x$ - und  $y$ -Richtung. Da die Gewichtskraft eine Volumenkraft ist, kann sie als Einzelkraft im SP angegeben werden und ist unabhängig vom Kraftschluss zwischen Fahrbahn und Reifen. Es wird die Annahme getroffen, dass die Fahrbahn in Querrichtung eben ist, sodass die Winkel nur von  $s$  und nicht von  $d$  abhängen, und somit  $\frac{\partial \theta}{\partial x} = \frac{\partial \phi}{\partial x} = \mathbf{0}$  ist.

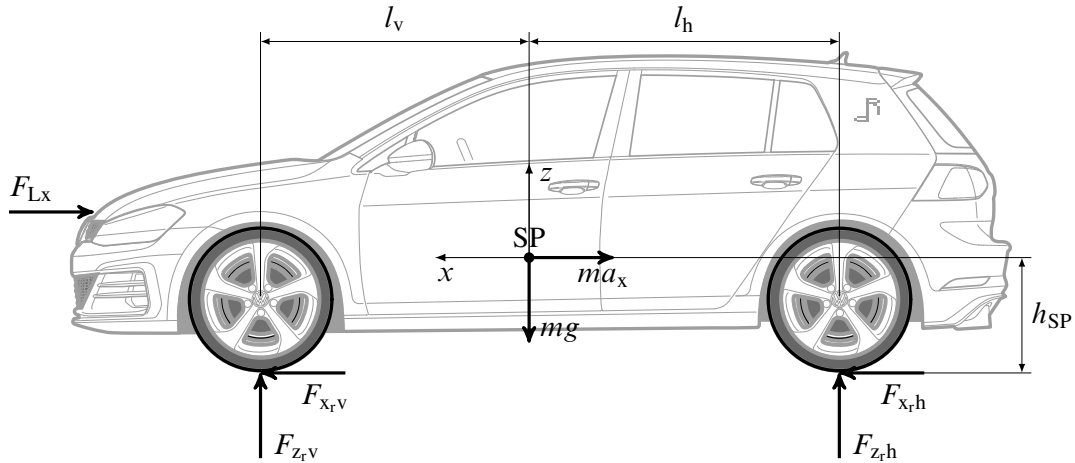
Ist eine Ideallinie gewünscht, deren Verlauf abhängig vom Höhenprofil gewählt wird, muss bei der Optimierung  $\Delta\psi_t \neq \text{const}$  berücksichtigt werden. Steht die Linienwahl nicht im Vordergrund, sondern lediglich die Tatsache, dass bspw. auf einer Steigung aus gleicher Motorleistung weniger Längsbeschleunigung resultiert, können  $\theta$  und  $\phi$  auf einer zuvor ohne Höhenprofil berechneten Linie angegeben werden, sodass  $\Delta\psi_t \approx 0$  ist und (3.42) über

$$F_{w,x} \approx F_{Lx} + mg \sin(\theta), \quad F_{w,y} \approx mg \sin(\phi) \quad (3.44)$$

approximiert werden kann. Achslastschwankungen aufgrund höherfrequenter Anregungen der Vertikaldynamik durch Bodenunebenheiten, Kuppen oder ähnliches, welche bspw. in [11] modelliert sind, werden hier nicht untersucht.

### 3.2.3 Normalkraft und Achslastverteilung

Unabhängig von einer Nickbewegung der Karosserie verändern sich beim Beschleunigen und Verzögern die dynamischen Achslasten, sobald die Höhe des Schwerpunkts  $h_{sp} > 0$  ist. Die Achslasten ergeben sich durch den Momentensatz um die Radaufstandspunkte. Anhand der Kräfte und



**Bild 3.6:** Die Schwerpunkthöhe  $h_{SP}$  ist der Hebelarm, über welchen die beschleunigte Fahrzeugmasse  $m$  die Achslastverteilung dynamisch beeinflusst.

Längen in Bild 3.6 lässt sich das Momentengleichgewicht in der Ebene bestimmen, wobei hier angenommen wird, dass das durch  $F_{Lx}$  entstehende Luftnickmoment  $M_{Ly}$  gering ist:

$$0 = F_{zrv}l - mgl_h + ma_x h_{SP} + \cancel{M_{Ly}}^0 \quad (3.45a)$$

$$0 = F_{zrh}l - mgl_v - ma_x h_{SP} - \cancel{M_{Ly}}^0 \quad (3.45b)$$

Damit ergibt sich aus der Schwerpunktlage die statische Massenverteilung auf die beiden Achsen:

$$m_v := m \frac{l_h}{l} \quad \text{und} \quad m_h := m \frac{l_v}{l} \quad \text{mit} \quad m = m_v + m_h, \quad l = l_v + l_h \quad (3.46)$$

In der Regel wird (3.46) verwendet, um aus der Massenverteilung (direkt messbar) die nicht direkt messbare Schwerpunktlage, also  $l_v$  und  $l_h$  zu berechnen, die das Fahrverhalten stark beeinflusst.

Bei den meisten Serienfahrzeugen entsteht (im Gegensatz zu Rennwagen) ein aerodynamischer Auftrieb, der sich in eine reine Auftriebskraft und ein Nickmoment aufteilen lässt. Mit dem Luftauftriebsbeiwert  $c_z$  (jeweils für vorne und hinten) und dem Luftmomentenbeiwert  $c_{My}$  um die  $y$ -Achse ergibt sich die resultierende Auftriebskraft [55, S. 56 ff.]

$$F_{Lz,v/h} = \frac{1}{2} \left( \frac{1}{2} c_{z,v/h} A \rho_L v^2 \pm \frac{1}{2} c_{My} A \rho_L v^2 \right) = \frac{1}{4} (c_{z,v/h} \pm c_{My}) A \rho_L v^2 \gtrless 0, \quad (3.47)$$

welche an beiden Achsen zu einer Reduktion der Aufstandskraft führt (obere Lesart von „ $\pm$ “ für Vorderachse). Jedoch ist sie i. A. niedrig und wird hier vernachlässigt.

Unter zusätzlicher Berücksichtigung der Fahrbahnneigungswinkel  $\theta$  und  $\phi$  ergeben sich aus (3.45) die dynamischen Achslasten. Dabei ist die SP-Lage konstant, da ihre geringe Verschiebung durch die Nickbewegung beim Einfedern vernachlässigt wird (vgl. [15, S. 34 ff.]).

$$F_{zr,v/h} = m_{v/h} g c_\theta c_\phi \mp \frac{h_{SP}}{l} m g \sin(\theta) \mp \frac{h_{SP}}{l} m a_x, \quad c_\theta c_\phi := \cos(\theta) \cos(\phi) \quad (3.48)$$

Die Längsbeschleunigung  $a_x$  ergibt sich aus dem Kräftegleichgewicht

$$ma_x = F_{x,v} \cos(\delta) - F_{y,v} \sin(\delta) + F_{x,h} - F_{Lx} - mg \sin(\theta), \quad (3.49)$$

allerdings führt sie in (3.48) zu einer komplexen Abhängigkeit von vielen Zuständen, da nicht nur die Reifenlängskräfte, sondern auch die vordere Reifenquerkraft  $F_{y,v}$  eingeht, welche über die Reifenkennlinie von  $\delta$ ,  $v$ ,  $\beta$  und  $\dot{\psi}$  abhängt. In Anbetracht dessen, dass die Aufstandskraft  $F_{z,v/h}$  nicht nur in die Modellgleichungen, sondern gemäß (2.3) auf S. 15 auch in den Kammschen Kreis eingeht, bringt eine Approximation, die zu einer Reduktion der Variablenabhängigkeiten führt, einen immensen Nutzen bezüglich des Rechenaufwands. Da der Anteil der Querkraft  $F_{y,v} \sin(\delta)$  an der Längsbeschleunigung relativ gering ist, wird er vernachlässigt. Der Luftwiderstand  $F_{Lx}$  hat zwar durchaus wichtigen Einfluss auf die Längsbeschleunigung per se, aber die Auswirkung auf die dynamische Achslastverteilung (3.48) ist im Vergleich zu den anderen Größen unwesentlich. Zusätzlich wird  $F_{x,v} \cos(\delta)$  um  $\delta = 0$  linearisiert, sodass sich die Approximation

$$ma_x \approx F_{x,v} + F_{x,h} - mg \sin(\theta) = F_M + F_B - mg \sin(\theta) \quad (3.50)$$

ergibt. Die nichtlineare Abhängigkeit von 6 Variablen reduziert sich auf 2 lineare Abhängigkeiten. Eingesetzt in (3.48) folgt

$$F_{z,v/h} = m_{v/h} g c_\theta c_\phi \mp \frac{h_{SP}}{l} (F_M + F_B), \quad (3.51)$$

worin sich der Ausdruck  $mg \sin(\theta)$  weghebt.

Nach Bild 3.4 auf S. 33 ist die Auswirkung von  $F_z$  auf das Reifenverhalten nichtlinear. Nimmt man die linearisierte Gleichung (3.28), kann in (3.51) die Schwerpunkthöhe  $h_{SP}$  formal durch eine effektive Schwerpunkthöhe

$$\tilde{h}_{SP} = (1 - k_{Fz}) h_{SP} \quad (3.52)$$

ersetzt werden. Für  $k_{Fz} = 0,3$  beträgt sie dann also nur 70 % der tatsächlichen Höhe.

### 3.2.4 Dynamische Bremskraftverteilung

Um eine maximale Verzögerung zu erzielen, müssen Vorder- und Hinterachse jeweils genau soviel Kraft übertragen, wie ihr jeweiliges Maximum zulässt. Da die Radlast die maximal übertragbare Reifenkraft beeinflusst, muss die Bremskraftverteilung  $k_{BV}$  in (3.37) auf S. 38 an die dynamische Achslastverteilung (3.51) so angepasst werden, dass die relative Umfangskraft  $F_{x,r}/F_{\max,x_r}$  an beiden Achsen gleich groß ist. Diese Auslegung ist die ideale Bremskraftverteilung.

Eine Verzögerung ( $-a_x$ ) bezogen auf die Erdbeschleunigung  $g$  bezeichnet man als *Abbremsung*. Sie wird allgemein mit dem Formelzeichen  $z$  symbolisiert, welches hier mit dem Index „B“ versehen wird, um eine Verwechslung mit der Hochachse zu vermeiden. Bei Geradeausfahrt ergibt sich aus dem dynamischen Gleichgewicht in Fzg-Längsrichtung [17, Kap. 3.1]

$$z_B := \frac{-a_x}{g} \Rightarrow z_B = \frac{-(F_{B,v} + F_{B,h})}{G}, \quad G := mg c_\theta c_\phi, \quad (3.53)$$

und somit die maximale Abbremsung

$$\hat{z}_B = \frac{-(\hat{F}_{B,v} + \hat{F}_{B,h})}{G} = \mu_x \frac{F_{z,v} + F_{z,h}}{G} = \mu_x. \quad (3.54)$$

Beim Verzögern ist die Motorkraft immer  $F_M = 0$  und die Bremskraft ist per Definition  $F_B \leq 0$ . Aus der Kraftschlussausnutzung (3.19) und der Achslast (3.51) folgt die maximale Bremskraft:

$$|\hat{F}_{B,v}(F_B)| = \mu_x F_{z,v}(F_B) \Big|_{F_M=0} = \mu_x \cdot \left( \frac{m_v}{m} G - \frac{h_{SP}}{l} F_B \right) \quad (3.55a)$$

$$|\hat{F}_{B,h}(F_B)| = \mu_x F_{z,h}(F_B) \Big|_{F_M=0} = \mu_x \cdot \left( \frac{m_h}{m} G + \frac{h_{SP}}{l} F_B \right) \quad (3.55b)$$

Für (3.37) ergibt sich aus (3.54) und (3.55a) die Bremskraftverteilung

$$k_{BV}(F_B) = \frac{\hat{F}_{B,v}(F_B)}{\hat{F}_{B,v}(F_B) + \hat{F}_{B,h}(F_B)} = \frac{\mu_x \cdot \left( \frac{m_v}{m} G - \frac{h_{SP}}{l} F_B \right)}{\mu_x G} = \underbrace{\frac{m_v}{m}}_{k_{BV,0}} - \underbrace{\frac{h_{SP}}{lmgc_\theta c_\phi}}_{k_{BV,p}} F_B, \quad (3.56)$$

welche zu einem konstanten Anteil  $k_{BV,0}$  und einem zu  $F_B$  proportionalen Anteil  $k_{BV,p}$  zusammengefasst wird. In (3.37) eingesetzt, erhält man also:

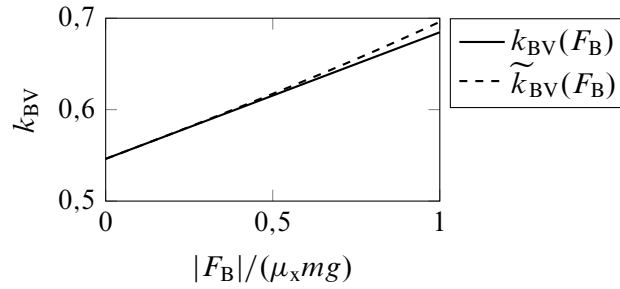
$$F_{x,v} = k_{MV} F_M + k_{BV,0} F_B - k_{BV,p} F_B^2 \quad (3.57a)$$

$$F_{x,h} = (1 - k_{MV}) F_M + (1 - k_{BV,0}) F_B + k_{BV,p} F_B^2 \quad (3.57b)$$

Für einen allgemeingültigen Wert bei  $h_{SP} = 0$  ist (3.56) allerdings ungeeignet, weil der resultierende Wert  $k_{BV} = k_{BV,0} = \text{const}$  typischerweise nur wenig größer als 0,5 ist, was in der Realität schon bei mittelstarker Verzögerung zu einem Überbremsen der Hinterachse führt. In dem Fall droht das Heck auszubrechen, wodurch das Fzg übersteuert und für einen ungeübten Fahrer schwer zu kontrollieren ist. Wird im umgekehrten Fall die Vorderachse überbremst, untersteuert das Fzg, was gerade auf Rennstrecken riskant ist, da es am Kurvenausgang zum Verlassen der Strecke führen kann.

Um den degressiven Achslasteinfluss (Bild 3.4 auf S. 33) zu berücksichtigen, sollte in (3.56) die effektive Schwerpunkthöhe  $\tilde{h}_{SP} = (1 - k_{Fz}) h_{SP}$  aus (3.52) eingesetzt werden. Allerdings wurde  $\tilde{h}_{SP}$  aus der Linearisierung des Achslasteinflusses (3.27) im Arbeitspunkt  $\Delta F_{z_r} = 0$  abgeleitet. Dieser Arbeitspunkt stellt sich bei  $F_B = 0$  ein und ist für eine Betrachtung des Bremsverhaltens möglicherweise ungeeignet. Daher wird im Folgenden die nicht linearisierte Bremskraftverteilung berechnet (hier für  $\theta = \phi = 0$ ):

$$\begin{aligned} \tilde{k}_{BV}(F_B) &= \frac{\tilde{F}_{z,v}(F_B)}{\tilde{F}_{z,v}(F_B) + \tilde{F}_{z,h}(F_B)} = \dots = \frac{m_h \left( m_v g - \frac{h_{SP}}{l} F_B \right) \left( m_v g + \frac{h_{SP}}{l} k_{Fz} F_B \right)}{m \left( m_v m_h g^2 - \left( \frac{h_{SP}}{l} \right)^2 k_{Fz} F_B^2 \right)} \quad (3.58) \\ &= \underbrace{\frac{m_v}{m} - (1 - k_{Fz}) \frac{h_{SP}}{lmg} F_B}_{=k_{BV} \text{ für } h_{SP} \rightarrow \tilde{h}_{SP}} + k_{Fz} \frac{h_{SP}^2 (m_v - m_h)}{l^2 g^2 m m_v m_h} F_B^2 + \mathcal{O}(F_B^3) \end{aligned}$$



**Bild 3.7:** Typischer Verlauf der Bremskraftverteilung in Abhängigkeit der Gesamtbremskraft.

Gl. (3.58) zeigt, dass der konstante und der lineare Teil identisch zu (3.56) mit  $\tilde{h}_{SP}$  sind, und der quadratische Anteil klein ist, da zum einen  $m_v \approx m_h$  ist und zum anderen die Potenz der Masse im Nenner zunimmt. Bild 3.7 zeigt einen typischen Verlauf für realistische Parameter eines Kompaktwagens mit  $k_{Fz} = 0,3$ .

Des Weiteren wird in (3.54) angenommen, dass die Gesamtbremskraft  $\hat{F}_B = \mu_x F_{zr} = \mu_x G$  ist. Da die maximale Umfangskraft bei einer Achslastverschiebung an der Vorderachse nicht im gleichen Maß zunimmt, wie sie an der Hinterachse abnimmt, ist auch diese Annahme nicht ganz korrekt. Aus (3.27) ergibt sich für Geradeausfahrt eine Abschätzung der tatsächlichen Maximal-Abbremsung  $\tilde{\hat{z}}_B$ :

$$\tilde{\hat{z}}_B \approx \mu_x \left. \frac{\tilde{F}_{zrv} + \tilde{F}_{zrh}}{G} \right|_{F_M + F_B = -mg} = \left( 1 - k_{Fz} \left( \frac{h_{SP}}{l} \right)^2 \frac{m^2}{m_v m_h} \right) \hat{z}_B. \quad (3.59)$$

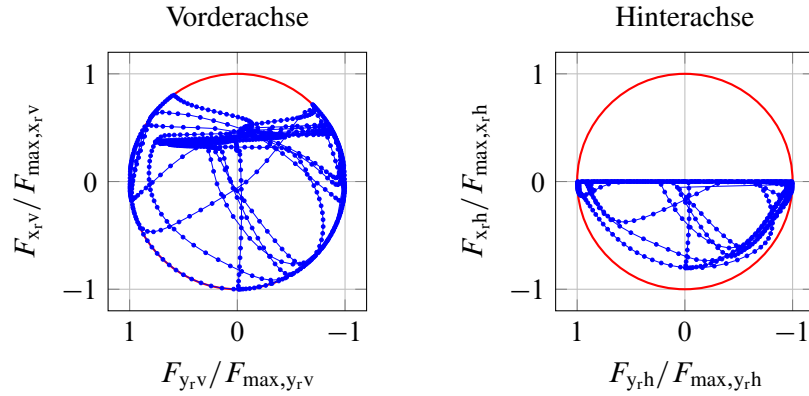
$\tilde{\hat{z}}_B$  ist typischerweise nur etwa 3–5 % kleiner als  $\hat{z}_B$ . Somit liegt man bei solchen Berechnungen in einer Genauigkeitsklasse, in der das Einspurmodell ohne Berücksichtigung von Fahrwerkseigenschaften und mit dem hier verwendeten Reifenmodell eh nicht mehr ausreichend genau ist. Verlässliche Aussagen darüber, ob ein gewisser Betriebspunkt gerade noch unter, exakt auf oder schon etwas über der physikalischen Grenze liegt, sind dann mit diesen Modellen nicht möglich.

### 3.2.5 Simulation zum Einfluss der Achslast auf die Kraftschlussgrenze

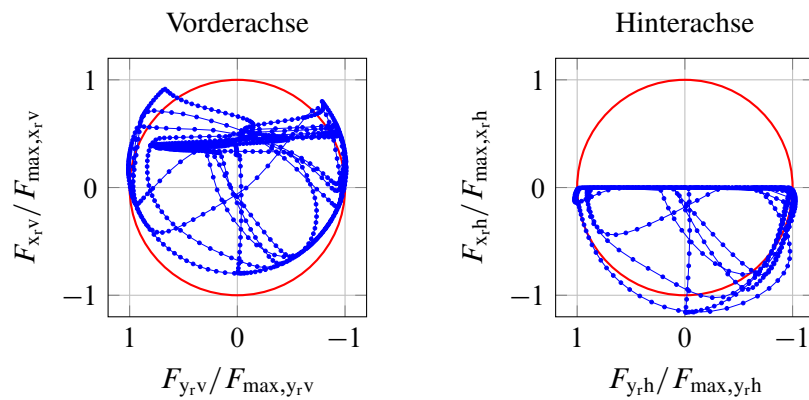
Im Folgenden wird gezeigt, wie sich die Achslastverteilung auf die Trajektorie auswirkt, und welchen Einfluss die in den letzten Abschnitten gewählten Vereinfachungen haben. Dafür wird für gewisse Konfigurationen eine Trajektorie berechnet und die Reifenkräfte bezogen auf die maximalen Reifenkräfte im Kammschen Kreis dargestellt. Als Strecke dient das *Autodrom Most* (Streckenverlauf ist in Bild 5.9 auf S. 90 zu sehen), weil dort alle relevanten Bereiche im Kammschen Kreis abgedeckt werden. Die Darstellung ist ähnlich zu dem bekannten GG-Diagramm für den SP, jedoch erlaubt sie zum einen eine separate Auswertung von Vorder- und Hinterachse und zum anderen ist jeder einzelne Punkt im Diagramm auf das Maximum seines Betriebspunkts und nicht auf ein einheitlich festgesetztes Maximum normiert.

Zunächst wird eine Trajektorie für  $h_{SP} = 0$  und somit zwangsläufig für eine konstante Bremskraftverteilung (BKV) geplant, deren relative Reifenkräfte in Bild 3.8(a) dargestellt sind. Da das Fzg





(a) Trajektorienplanung für  $h_{SP} = 0$  und  $k_{BV} = 0,6 = \text{const}$ , sodass die Maximalkraft  $F_{\max} = \mu F_{Z_r} = \text{const}$  ist.

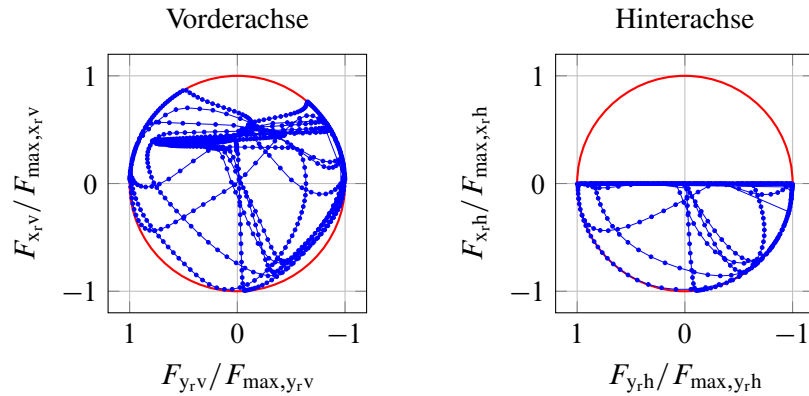


(b) Die Auswertung der Trajektorie aus (a) unter Berücksichtigung einer dynamischen Achslastverteilung  $F_{\max} = f(a_x)$  gemäß (3.51) für eine Schwerpunkthöhe  $\tilde{h}_{SP} > 0$  gemäß (3.52) zeigt an beiden Achsen eine signifikante Überschreitung der Maximalkraft um bis zu 16 %.

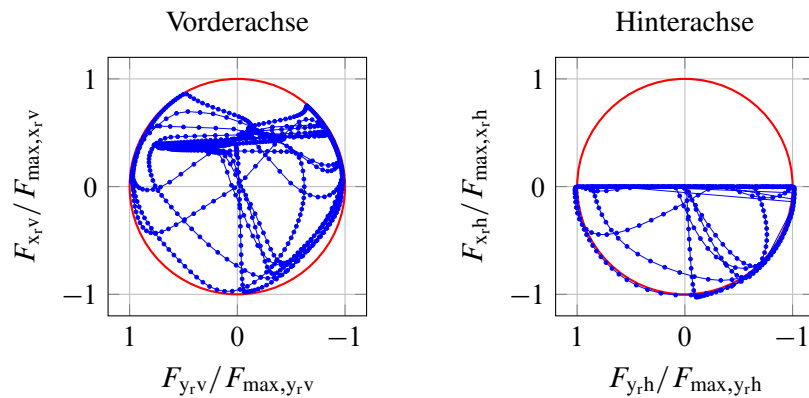
**Bild 3.8:** Einfluss der Achslastverteilung beim Beschleunigen und Verzögern.

frontgetrieben ist, können an der Hinterachse keine positiven Längskräfte entstehen. Aufgrund der begrenzten Motorleistung kann in positiver Längsrichtung die Kraftschlussgrenze nicht immer erreicht werden, wobei man erkennen kann, dass beim Herausbeschleunigen aus Kurven, wo die Geschwindigkeit tendenziell niedriger ist, die verfügbare Längskraft gemäß (3.41) auf S. 39 tendenziell höher ist als auf Geraden.

Nach (3.56) ergäbe sich für die gewählten Fzg-Parameter bei  $h_{SP} = 0$  eine hecklastigere (niedrigere) BKV als die hier gewählte von  $k_{BV} = 0,6$ . Allerdings wurde bereits gesagt, dass (3.56) für  $h_{SP} = 0$  ungeeignet ist, was im Folgenden demonstriert wird. In der Planung führt die (mutmaßlich frontlastige) BKV dazu, dass die Vorderachse die Kraftschlussgrenze erreicht, während die Hinterachse noch Potential hat. Nun wird die geplante Trajektorie nachträglich unter Berücksichtigung einer dynamischen Achslastverteilung ausgewertet, indem die Maximalkräfte  $F_{\max,x_r}$  und  $F_{\max,y_r}$  mit (3.51) neu berechnet werden. Man kann in Bild 3.8(b) sehen, dass die Hinterachse schon mit  $k_{BV} = 0,6$  stark überbremsst wird. Zudem kann man sehen, dass durch die Entlastung der Vorderachse beim Beschleunigen die Kraftschlussgrenze auch in positiver Richtung überschritten



(a) Trajektorienplanung mit dynamischer Achslastverteilung (3.51) und dynamischer Bremskraftverteilung gemäß (3.56) respektive einer effektiven Schwerpunkthöhe  $\tilde{h}_{SP} > 0$  gemäß (3.52).



(b) Die Auswertung der Trajektorie aus (a) unter Berücksichtigung der nicht approximierten dynamischen Achslastverteilung gemäß (3.48) respektive (3.49) zeigt lediglich an der Hinterachse eine geringe Überschreitung der Maximalkraft um bis zu 3 %.

**Bild 3.9:** Einfluss der Approximation bei der Berechnung der Achslastverteilung nach (3.51).

wird. Daher ist die Modellierung der dynamischen (quasi-statischen) Achslast- und Bremskraftverteilung im Grenzbereich wichtig. Allerdings steigt dadurch die Rechenzeit um ca. 7 %.

Eine Approximation, die zwar das Optimierungsproblem vereinfacht, aber möglicherweise die Trajektorie zu stark beeinflussen könnte, besteht in der Approximation der Achslast (3.48) durch (3.51) bzw. der Längsbeschleunigung (3.49) durch (3.50). Um dies zu untersuchen, wird eine Trajektorie mit approximierter Achslastverteilung (und dynamischer BKV) berechnet. Ihr Kraftverlauf ist in Bild 3.9(a) abgebildet.

Im Gegensatz zu Bild 3.8 erreichen Vorder- und Hinterachse beim reinen Verzögern gleichzeitig ihre Kraftschlussgrenze. Da die BKV aufgrund jener Approximation nur für Geradeausfahrt ideal ist, wird beim kombinierten Bremsen und Lenken etwas Potential an der Vorderachse verschenkt, was aber nicht sicherheitskritisch und auch für die Rundenzeit vernachlässigbar ist.

In Bild 3.9(b) wird die Trajektorie nun ohne die Approximation ausgewertet, indem wiederum die Maximalkräfte  $F_{\max, x_r}$  und  $F_{\max, y_r}$  mit (3.48) und (3.49) neu berechnet werden. Bei maximaler Verzögerung aus hoher Geschwindigkeit (in Most beim Anbremsen am Ende der langen Geraden vor der Schikane) zeigt sich eine leichte Überschreitung (3 %) an der Hinterachse, was auf die Vernachlässigung des Luftwiderstands zurückzuführen ist. Denn dadurch ist die approximierte Längsbeschleunigung größer als die tatsächliche oder – anders formuliert – die approximierte Verzögerung geringer als die tatsächliche, sodass die Approximation der hinteren Achslast geringer als die tatsächliche Achslast ist. Daher wird die Hinterachse überbremst, während die Vorderachse aufgrund der höheren Achslast ihr Limit noch nicht erreicht. Eine noch kleinere Überschreitung der Kraftschlussgrenze von 2 % entsteht beim kombinierten Verzögern und Lenken, was durch die Vernachlässigung des Lenkwinkels resultiert. An der Vorderachse liegt keine Überschreitung vor. Der Einfluss der gewählten Approximation fällt somit sicher geringer aus, als die Auswirkungen, die sowohl durch den Detaillierungsgrad des gesamten Fzg-Modells (Fahrwerkeinflüsse, Reifenzustand, Aerodynamik etc.) als auch durch die Genauigkeit der Fahrbahnmodellierung (Unebenheiten, Temperaturabhängigkeit etc.) entstehen. Daher ist diese Approximation durchaus zweckmäßig.

### 3.3 Zusammenfassung

Durch die Transformation des Optimierungsproblems vom Zeit- in den Weg-Bereich kann das zeitoptimale Gütemaß mit festen Integrationsgrenzen formuliert werden. Dazu wird die Fahrzeugpose relativ zu einer Referenzlinie angegeben, welche den Streckenverlauf in Form ihrer Krümmung abhängig von der Bogenlänge beschreibt. Die Referenzlinie bildet das Koordinatensystem für die gesamte Planung, sodass in der Planung keine globalen Koordinaten vorkommen.

Die Fahrzeugdynamik wird mit einem nichtlinearen Einspurmodell beschrieben, dessen DGLs ebenfalls von der Bogenlänge der Referenzlinie abhängen. Das Verhalten der Reifen hat einen großen Einfluss auf die Fahrzeugdynamik. Für die Modellierung der Achsquerkraft wird die *Magic Formula* nach PACEJKA [59] verwendet, weil sie mehrfach stetig differenzierbar ist, sich verhältnismäßig einfach parametrieren lässt und das Reifenverhalten bis zum Grenzbereich hinreichend genau beschreibt. Das Einspurmodell und insbesondere das Reifenmodell wird so erweitert, dass eine beschleunigungsabhängige Achslastverteilung durch die Massenträgheit berücksichtigt wird. Dies führt zu einer ebenfalls beschleunigungsabhängigen Bremskraftverteilung auf Vorder- und Hinterachse. Anhand von Simulationen werden die Auswirkungen der Achslast- und Bremskraftverteilung auf die Kraftschlussgrenze gezeigt.

## 4 Methoden der nichtlinearen Optimierung

Eine Trajektorienplanung soll unter unendlich vielen Trajektorien die optimale Trajektorie finden. Welche Trajektorie optimal ist, wird über ein Gütemaß definiert, welches jeder Trajektorie einen skalaren Zahlenwert zuordnet. Für eine praktische Umsetzung können nicht unendlich viele Trajektorien beurteilt werden, sondern es muss eine Methodik angewendet werden, die möglichst schnell die optimale Trajektorie berechnet. Eine solche Methodik heißt Optimierung.

Da in der vorliegenden Arbeit die Trajektorie über den Weg parametrisiert ist, wird im Folgenden von einem unabhängigen Parameter  $s$  ausgegangen, der sich im Übrigen genauso wie ein unabhängiger Zeitparameter  $t$  verhält. Die Zuordnung des Gütemaßes zur Trajektorie soll den gesamten Definitionsbereich  $s \in [0, S_{\text{end}}]$  berücksichtigen, wobei der Bereich o. B. d. A. bei  $s = 0$  beginne. Eine Gütefunktion  $\varphi$  ordnet jedem Trajektorienpunkt  $\mathbf{x}(s^*) \in \mathbb{R}^n$  eine reelle Zahl zu. Das Gütefunktional ist das Integral über  $\varphi$  zuzüglich des MAYER-Terms  $\vartheta$  zur Bewertung des Endpunkts. Dieses Gütefunktional  $J$  ordnet also der gesamten Trajektorie  $\mathbf{x}(s)$  aus dem  $n$ -dimensionalen Funktionenraum eine reelle Zahl zu:

$$\min \quad J = \int_{s=0}^{S_{\text{end}}} \varphi(\mathbf{x}(s)) \, ds + \vartheta(\mathbf{x}(S_{\text{end}})) \quad (4.1)$$

Meistens unterliegt das Optimierungsproblem Nebenbedingungen (NB), sodass man

$$\mathbf{x} \in \Omega \quad \text{mit} \quad \Omega = \{\mathbf{x} \mid \mathbf{g}(\mathbf{x}) = \mathbf{0}; \mathbf{h}(\mathbf{x}) \geq \mathbf{0}\} \quad (4.2)$$

fordert.  $\mathbf{g}$  bezeichnet man als Gleichungsnebenbedingung (GNB) und  $\mathbf{h}$  als Ungleichungsnebenbedingung (UNB). Darüber hinaus sei der Anfangszustand  $\mathbf{x}(0) = \mathbf{x}_0$  gegeben.

Die numerischen Verfahren zur Lösung des Optimierungsproblems lassen sich in *direkte* und *indirekte* Verfahren klassifizieren. Bei direkten Verfahren wird das dynamische Optimierungsproblem diskretisiert und so in ein statisches Ersatzproblem überführt. Indirekte Verfahren lösen das Randwertproblem, das sich aus den sogenannten notwendigen Bedingungen 1. Ordnung für ein Optimum ergibt, welche in den folgenden Abschnitten hergeleitet werden. So können sie zwar genauere Lösungen berechnen, jedoch ist ihre Initialisierung oft schwieriger, weil schon bei der Initialisierung bekannt sein muss, in welcher Reihenfolge die UNB aktiv werden [60, S. 393]. Direkte Verfahren wurden früher weniger eingesetzt, weil die entstehenden Ersatzprobleme schnell große Dimensionen erreichen und die Kapazität der damaligen Rechner überstiegen [60, S. 387 f.]. Bei der hier entwickelten Trajektorienplanung ist die Trajektorie aufgrund der Zeitoptimalität permanent am Rand einer UNB. Allein diese Tatsache sowie auch die unproblematischere Verwendung ungünstiger Startlösungen und die höhere Verfügbarkeit von nicht kommerziellen Solvern sind die Argumente dafür, ein direktes Verfahren einzusetzen.

## 4.1 Diskretisierung eines dynamischen Optimierungsproblems

In der dynamischen Optimierung entstehen GNB oft durch die Differentialgleichungen eines dynamischen Systems mit den Eingängen  $\mathbf{u}$  und den Zuständen  $\mathbf{x}$ . Beide hängen in unserem Fall vom Weg  $s$  ab, was aber in den folgenden Formeln zur besseren Übersicht weggelassen wird. Das System sei also beschrieben durch

$$\mathbf{x}' := \frac{d\mathbf{x}}{ds} = \mathbf{f}_{\text{sys}}(\mathbf{x}, \mathbf{u}). \quad (4.3)$$

Das allgemeine Optimierungsproblem (4.1) – (4.2) wird nun als spezielles Optimierungsproblem mit einer DGL als GNB formuliert:

$$\min \quad J = \int_{s=0}^{s_{\text{end}}} \varphi(\mathbf{x}, \mathbf{u}) ds + \vartheta(\mathbf{x}_{\text{end}}) \quad (4.4a)$$

$$\text{u. B. v.} \quad \mathbf{x}' = \mathbf{f}_{\text{sys}}(\mathbf{x}, \mathbf{u}), \quad \mathbf{x}(0) = \mathbf{x}_0, \quad (4.4b)$$

$$\mathbf{h}(\mathbf{x}, \mathbf{u}) \geq \mathbf{0} \quad (4.4c)$$

Um das Problem mit den Methoden der statischen Optimierung lösen zu können, diskretisiert man es. Man unterscheidet dabei zwischen *Teildiskretisierung* („direkt sequentiell“) und *Volldiskretisierung* („direkt simultan“) [33, Kap. 6.2].

Bei der **Teildiskretisierung** wird nur der Stellgrößenverlauf an den Stellen  $s = s_1, \dots, s_N$  diskretisiert, wobei  $s_1 = 0$  und  $s_{N+1} = s_{\text{end}}$  ist.<sup>1</sup> So entstehen  $N$  Segmente und  $(N+1)$  Diskretisierungspunkte (DP). Für jedes Segment gibt es eine Optimierungsvariable pro Stellgröße. Die Trajektorie ist anhand der Stellgrößen durch die Systemgleichung (4.4b) eindeutig bestimmt, sodass die Zustände in Abhängigkeit der Stellgrößen formuliert werden können. Die Trajektorie selbst wird durch eine *externe* Simulation berechnet. Da Optimierung und Simulation nacheinander ablaufen, heißt das Verfahren auch „*direkt sequentiell*“.

Bei der **Volldiskretisierung** werden Stellgrößenverlauf *und* Zustandsverlauf diskretisiert. Der Optimierungsvektor besteht also aus den Stellgrößen für alle  $N$  Segmente und den Zuständen für alle  $(N+1)$  DP. Die *Differentialgleichung* (4.4b) wird in Form einer *Differenzengleichung* als GNB in das Optimierungsproblem aufgenommen und somit *simultan* integriert. Da in jeder Iteration die gesamte Trajektorie verbessert wird, bietet dieses Verfahren eine Art Parallelisierung auf algorithmischer Ebene.

Vorteile der Teildiskretisierung sind die minimale Anzahl an Optimierungsvariablen und die Flexibilität, NB auch zwischen den DP auswerten zu können, während die NB bei der Volldiskretisierung nur an den DP berücksichtigt werden. Das gilt auch für die Systemgleichungen. Die Teildiskretisierung ist besonders gut geeignet, wenn die Lösung der System-DGLs analytisch erfolgen kann, was aber bei nichtlinearen Systemen i. A. unmöglich ist. Bei gradientenbasierten

<sup>1</sup>Der Zählindex ist einsbasiert, vgl. Notation des Zählindex auf S. 8.

Optimierungsverfahren müssen die Ableitungen des Gütemaßes und der NB nach den Optimierungsvariablen bekannt sein. Bei komplexen Zusammenhängen sind die Ableitungen nach den Zuständen bereits diffizil. Diese jedoch wie bei einer Sensitivitätsanalyse nach den Stellgrößen zu berechnen, wie es die Teildiskretisierung erfordert, kann extrem aufwendig sein. Ein weiterer großer Nachteil der Teildiskretisierung ist, dass man eine gute Startlösung benötigt, da anderenfalls die Optimierung abbricht. Wenn z. B. die Stellgröße in der Startlösung dazu führt, dass das Fzg in der Simulation ins Schleudern gerät oder die Strecke verlässt, kann der Rest der Strecke nicht mehr simuliert werden. Die Volldiskretisierung kann hingegen mit jeder Startlösung starten.

In beiden Fällen erhält man ein statisches Optimierungsproblem der Form

$$\min_{\mathbf{x}_{\text{opt}}} J = f(\mathbf{x}_{\text{opt}}) \quad (4.5a)$$

$$\text{u. B. v. } \mathbf{g}(\mathbf{x}_{\text{opt}}) = \mathbf{0}, \quad (4.5b)$$

$$\mathbf{h}(\mathbf{x}_{\text{opt}}) \geq \mathbf{0} \quad (4.5c)$$

$$\text{mit } \mathbf{x}_{\text{opt}} = \begin{cases} \begin{bmatrix} \mathbf{u}_1^\top & \cdots & \mathbf{u}_N^\top \end{bmatrix}^\top, & \text{Teildiskretisierung} \\ \begin{bmatrix} \mathbf{u}_1^\top & \cdots & \mathbf{u}_N^\top & \mathbf{x}_1^\top & \cdots & \mathbf{x}_{N+1}^\top \end{bmatrix}^\top, & \text{Volldiskretisierung} \end{cases} \quad (4.5d)$$

$$\text{und } f(\mathbf{x}_{\text{opt}}) = \sum_{k=1}^N \varphi(\mathbf{x}_k, \mathbf{u}_k) \Delta s_k + \vartheta(\mathbf{x}_{N+1}), \quad \Delta s_k := s_{k+1} - s_k \quad (4.5e)$$

Dabei ist die Sortierung der Variablen in  $\mathbf{x}_{\text{opt}}$  beliebig wählbar. In den UNB  $\mathbf{h}$  in (4.5c) sind auch sogenannte Box-Restriktionen (engl. box constraints) der Form

$$\mathbf{x}_l \leq \mathbf{x}_{\text{opt}} \leq \mathbf{x}_u \quad (4.6)$$

enthalten, welche aufgrund ihrer einfachen Form oft separat behandelt werden. Ein allgemeines statisches Optimierungsproblem gemäß (4.5a) – (4.5c) lässt sich mit diversen Methoden lösen. Die für diese Arbeit wichtigsten Aspekte werden im Folgenden dargelegt, wobei sich ausschließlich auf gradientenbasierte Verfahren beschränkt wird, mit denen ein lokales Minimum gefunden werden kann.

## 4.2 Statische Optimierung ohne Nebenbedingungen

Zuerst soll das Optimierungsproblem (4.5a) für  $\mathbf{x}_{\text{opt}} \in \mathbb{R}^{N_{\text{opt}}}$  ohne NB betrachtet werden. Im Minimum gilt die *notwendige Bedingung 1. Ordnung*

$$\nabla f(\mathbf{x}_{\text{opt}}^*) = \mathbf{0}, \quad (4.7)$$

welche aber von allen stationären Punkten erfüllt wird, also auch von Maxima und Sattelpunkten [60, Kap. 4.1]. Daher fordert die strengere *notwendige Bedingung 2. Ordnung*, dass zusätzlich die HESSE-Matrix positiv semidefinit ist:

$$\nabla f(\mathbf{x}_{\text{opt}}^*) = \mathbf{0} \quad \text{und} \quad \nabla^2 f(\mathbf{x}_{\text{opt}}^*) \succeq 0. \quad (4.8)$$

Die *hinreichende Bedingung* für ein striktes lokales Minimum lautet:

$$\nabla f(\mathbf{x}_{\text{opt}}^*) = \mathbf{0} \quad \text{und} \quad \nabla^2 f(\mathbf{x}_{\text{opt}}^*) > 0. \quad (4.9)$$

Ist die HESSE-Matrix für alle  $\mathbf{x}_{\text{opt}}$  positiv definit, erfüllt jeder Punkt, der (4.7) erfüllt, auch die hinreichende Bedingung (4.9), sodass jeder stationäre Punkt ein globales Minimum ist. Dies trifft insbesondere auf quadratische Optimierungsprobleme (QP) zu.

Um den Punkt  $\mathbf{x}_{\text{opt}}^*$  ausgehend von einem Startpunkt  $\mathbf{x}_{\text{opt}}^{(0)}$  beim Iterationsindex  $i = 0$  zu finden, verfolgen numerische Verfahren üblicherweise einen iterativen Algorithmus nach folgender Struktur [60, Kap. 4.2]:

1. Bestimmung einer Suchrichtung  $\mathbf{p}^{(i)}$ , sodass die Abstiegsbedingung

$$f(\mathbf{x}_{\text{opt}}^{(i+1)}) = f(\mathbf{x}_{\text{opt}}^{(i)} + \alpha^{(i)} \mathbf{p}^{(i)}) < f(\mathbf{x}_{\text{opt}}^{(i)}) \quad (4.10)$$

für ein  $\alpha^{(i)} > 0$  erfüllt ist. Sie wird bei einem hinreichend kleinen  $\alpha$  erfüllt, sobald Gradient und Suchrichtung einen stumpfen Winkel bilden [58, S. 21 f.], also

$$\nabla^\top f(\mathbf{x}_{\text{opt}}^{(i)}) \mathbf{p}^{(i)} < 0. \quad (4.11)$$

2. Bestimmung der skalaren Schrittweite  $\alpha^{(i)} > 0$  (*Liniensuche*) und Berechnung des nächsten Punkts

$$\mathbf{x}_{\text{opt}}^{(i+1)} = \mathbf{x}_{\text{opt}}^{(i)} + \alpha^{(i)} \mathbf{p}^{(i)}. \quad (4.12)$$

3. Überprüfung des Abbruchkriteriums und ggf. Wiederholung des Algorithmus für  $i \leftarrow i + 1$ .

Im Folgenden wird der Iterationsindex  $i$  zur besseren Übersicht weggelassen, wenn sich die Gleichungen auf eine einzelne Iteration beziehen.

### 4.2.1 Bestimmung der Suchrichtung

Die naheliegendste Methode ist, in Richtung des **steilsten Abstiegs** zu laufen, welche durch den Gradienten  $\hat{\mathbf{p}} = -\nabla f(\mathbf{x}_{\text{opt}})$  gegeben ist und somit die Abstiegsbedingung (4.11) erfüllt, da  $\forall \mathbf{x}_{\text{opt}} \neq \mathbf{x}_{\text{opt}}^* : -\nabla^\top f(\mathbf{x}_{\text{opt}}) \nabla f(\mathbf{x}_{\text{opt}}) < 0$  ist. Diese Methode hat zwar den geringsten Rechenaufwand, da lediglich der Gradient zu berechnen ist, jedoch kann es bei komplexen Problemen zu schleichender Konvergenz kommen [58, S. 21].

Eine der wichtigsten Methoden ist das **NEWTON-Verfahren**, das zu einer deutlich besseren Konvergenz führt. Hierbei wird  $f$  an der Stelle  $\mathbf{x}_{\text{opt}}$  in eine TAYLOR-Reihe 2. Ordnung entwickelt, um eine Approximation für  $f(\mathbf{x}_{\text{opt}} + \mathbf{p})$  zu erhalten [58, S. 22]:

$$f(\mathbf{x}_{\text{opt}} + \mathbf{p}) \approx f(\mathbf{x}_{\text{opt}}) + \mathbf{p}^\top \nabla f(\mathbf{x}_{\text{opt}}) + \frac{1}{2} \mathbf{p}^\top \nabla^2 f(\mathbf{x}_{\text{opt}}) \mathbf{p} =: \mathcal{F}(\mathbf{p}) \quad (4.13)$$

Wenn die HESSE-Matrix  $\nabla^2 f$  hinreichend glatt ist, bietet  $\mathcal{F}(\mathbf{p})$  eine gute Approximation. Grundsätzlich muss die Gütefunktion hierfür natürlich zweimal stetig differenzierbar sein, damit sie sich in akzeptabler Weise vorhersagbar verhält, und die Algorithmen eine sinnvolle Suchrichtung bestimmen können. Unter der Annahme, dass die HESSE-Matrix positiv definit ist, was nach (4.9) zumindest in der Nähe der Lösung der Fall ist, findet man die NEWTON-Richtung  $\mathbf{p}$ , indem man analog zu (4.7) den Gradienten der TAYLOR-Approximation, also  $\nabla \mathcal{F}(\mathbf{p})$  zu null setzt:

$$\nabla \mathcal{F}(\mathbf{p}) = \nabla f + \nabla^2 f \mathbf{p} \stackrel{!}{=} \mathbf{0} \quad \Rightarrow \quad \mathbf{p} = -(\nabla^2 f)^{-1} \nabla f \quad (4.14)$$

Für eine positiv definite HESSE-Matrix ist die Abstiegsbedingung (4.11) damit erfüllt:

$$\nabla^\top f \mathbf{p} = -\nabla^\top f (\nabla^2 f)^{-1} \nabla f < 0 \quad (4.15)$$

Ist sie nicht positiv definit, ist  $\mathbf{p}$  möglicherweise keine geeignete Suchrichtung. In dem Fall kann die HESSE-Matrix modifiziert werden, indem bspw. negative Eigenwerte durch einen kleinen positiven Wert, der etwas größer als die Maschinengenauigkeit ist, ersetzt werden [58, S. 49 ff.]. Wird bei der Liniensuche die Schrittweite  $\alpha < 1$  gewählt, weil ein ganzer Schritt nicht zu der notwendigen Reduktion von  $f$  führen würde, spricht man von einem *gedämpften* NEWTON-Verfahren.

Für die numerische Berechnung der Suchrichtung wird (4.14) nicht über die inverse HESSE-Matrix, sondern über das lineare Gleichungssystem (LGS)

$$\nabla^2 f \mathbf{p} = -\nabla f \quad (4.16)$$

bestimmt. Jedoch kann allein schon die explizite Berechnung der HESSE-Matrix ein „mühseliger, fehleranfälliger und aufwendiger Prozess“ [58, S. 23] sein. Der Aufwand kann jedoch einen großen Nutzen für die Konvergenz bringen, wie sich in diesem Projekt gezeigt hat. Zum Lösen des LGS eignen sich z. B. die Algorithmen der HARWELL SUBROUTINE LIBRARY (HSL) [78], die in FORTRAN programmiert sind. Bei hochdimensionalen Problemen mit komplexen NB benötigt das Bestimmen der Suchrichtung oft den größten Anteil der gesamten Rechenzeit, sodass man unbedingt einen geeigneten Solver hierfür einsetzen sollte.

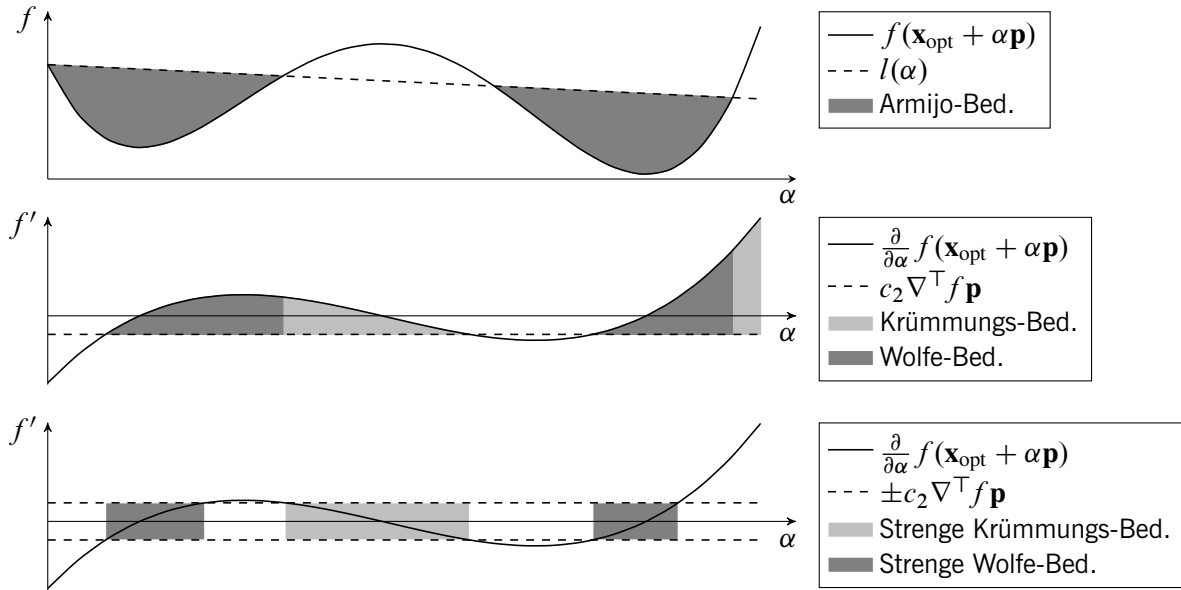
Um die HESSE-Matrix nicht explizit berechnen zu müssen, kann sie über ein **Quasi-NEWTON-Verfahren** approximiert werden [58, S. 23 ff., 135 ff.]. Dabei geht man davon aus, dass in der Nähe der Lösung der Term (4.14) durch den Ausdruck  $\nabla^2 f \mathbf{p}$  dominiert wird, da der Gradient gegen null geht.

Mit  $\mathbf{p}^{(i)} = \mathbf{x}_{\text{opt}}^{(i+1)} - \mathbf{x}_{\text{opt}}^{(i)}$  und  $\mathcal{F}(\mathbf{p}^{(i)}) \approx f(\mathbf{x}_{\text{opt}}^{(i+1)})$  folgt aus (4.14)

$$\nabla^2 f^{(i)} \mathbf{p}^{(i)} \approx \nabla f^{(i+1)} - \nabla f^{(i)}, \quad (4.17)$$

wobei  $f^{(i)} := f(\mathbf{x}_{\text{opt}}^{(i)})$  ist. Ausgehend von dieser Approximation existieren verschiedene iterative Berechnungsvorschriften, wie die *BFGS*-Formel, um die HESSE-Matrix oder direkt die inverse HESSE-Matrix durch eine symmetrische, positiv definite Matrix zu approximieren. Im ersten Iterationschritt dient die Richtung des steilsten Abstiegs als Suchrichtung und für die HESSE-Matrix wird die Einheitsmatrix angesetzt [60, S. 49].





**Bild 4.1: Oben:** Die ARMIJO-Bedingung fordert eine Reduzierung des Funktionswerts.

**Mitte:** Die Krümmungsbedingung beschränkt den Gradienten nach unten und führt zusammen mit der ARMIJO-Bedingung zu den WOLFE-Bedingungen.

**Unten:** Die strenge Krümmungsbedingung gibt auch eine obere Grenze des Gradienten vor.

Auf weitere Verfahren, wie das *Konjugierte Gradientenverfahren* und das *Trust-Region-Verfahren*, wird hier nicht eingegangen. In dieser Arbeit wird das NEWTON-Verfahren verwendet, weil es die beste Konvergenz bietet. Zudem ist die NEWTON-Richtung aufgrund der Approximation 2. Ordnung unempfindlich gegenüber einer schlechten Konditionierung der Gütefunktion, sodass auf eine lineare Skalierung sogar ganz ohne Einbußen verzichtet werden kann [60, S. 56], [58, S. 26 f.].

## 4.2.2 Liniensuche

Ziel der Liniensuche ist es, diejenige Schrittweite  $\alpha > 0$  zu finden, die die Funktion  $f$  am meisten reduziert. Für eine **exakte Liniensuche** muss

$$\begin{aligned} \alpha &= \arg \min_{\alpha} f(\mathbf{x}_{\text{opt}} + \alpha \mathbf{p}) \\ \Rightarrow 0 &\stackrel{!}{=} \frac{\partial f(\mathbf{x}_{\text{opt}} + \alpha \mathbf{p})}{\partial \alpha} = \nabla^\top f(\mathbf{x}_{\text{opt}} + \alpha \mathbf{p}) \frac{\partial(\mathbf{x}_{\text{opt}} + \alpha \mathbf{p})}{\partial \alpha} = \nabla^\top f(\mathbf{x}_{\text{opt}} + \alpha \mathbf{p}) \mathbf{p} \end{aligned} \quad (4.18)$$

sein. Die Auswertung von (4.18) bringt oft einen zu hohen numerischen Aufwand mit sich.

Eine weit verbreitete **inexakte Liniensuche** ist die Auswertung der **WOLFE-Bedingungen** [58, S. 33 ff.]. Zunächst fordert man, dass  $f^{(i+1)}$  kleiner ist, als der Wert einer Funktion  $l \propto \alpha$ :

$$f(\mathbf{x}_{\text{opt}} + \alpha \mathbf{p}) \leq \underbrace{f(\mathbf{x}_{\text{opt}}) + c_1 \alpha \nabla^\top f \mathbf{p}}_{l(\alpha)}, \quad c_1 \in (0, 1) \quad (4.19)$$

Die ARMIJO-Bedingung (4.19) fordert, dass die Reduktion in  $f$  sowohl zu  $\alpha$  als auch zur Richtungsableitung  $\nabla^\top f \mathbf{p}$  proportional ist. In Bild 4.1 (oben) sind die Bereiche, die dieser Bedingung

genügen, grau hinterlegt. Da zwangsläufig alle  $\alpha \approx 0$  die Bedingung erfüllen, was zu einer unnötig kleinen Schrittweite führen könnte, fordert man zusätzlich, dass die Steigung an der Stelle  $(\mathbf{x}_{\text{opt}} + \alpha \mathbf{p})$  um einen gewissen Faktor  $c_2$  größer sein muss, als die Steigung bei  $\alpha = 0$ :

$$\nabla^\top f(\mathbf{x}_{\text{opt}} + \alpha \mathbf{p}) \mathbf{p} \geq c_2 \nabla^\top f \mathbf{p}, \quad c_2 \in (c_1, 1) \quad (4.20)$$

Die ARMIJO-Bedingung (4.19) und die Krümmungsbedingung (4.20) ergeben gemeinsam die WOLFE-Bedingungen. Bild 4.1 (Mitte) zeigt  $f'(\alpha)$  und die Bereiche, in denen die Krümmungs- und WOLFE-Bedingungen erfüllt sind. Wie man darin erkennen kann, ist (4.20) sinnvoll, weil die Steigung  $f'(\alpha)$  zu Beginn immer negativ ist und man so einen Indikator dafür hat, dass man  $f$  signifikant reduzieren kann, wenn man in die gewählte Richtung geht [58, S. 34]. Da ein Punkt zwar den WOLFE-Bedingungen genügen, aber trotzdem weit vom Optimum entfernt sein kann, fordern die *strengen* WOLFE-Bedingungen, dass die Steigung auch nach oben begrenzt wird:

$$|\nabla^\top f(\mathbf{x}_{\text{opt}} + \alpha \mathbf{p}) \mathbf{p}| \leq c_2 |\nabla^\top f \mathbf{p}| \quad (4.21)$$

So landet man in der Nähe eines stationären Punkts von  $f$ , wie Bild 4.1 (unten) zeigt.

Die WOLFE-Bedingungen sagen zwar, welche Schrittweiten geeignet sind, sie sagen aber nicht, wie man sie findet oder gar berechnet. Viele Algorithmen finden ein geeignetes  $\alpha$  durch geschicktes Ausprobieren, wobei es wichtig ist, die meist aufwendige Berechnung des Gradienten möglichst selten durchführen zu müssen. Wird die Suchrichtung mittels NEWTON-Verfahren bestimmt, beginnt man mit der Schrittweite  $\alpha_0 = 1$ . Genügt sie nicht der ARMIJO-Bedingung (4.19), ist sie zu groß, und es muss im Intervall  $(0, \alpha_0)$  eine geeignete Schrittweite geben (s. Bild 4.1) [58, Kap. 3.5]. Für die Berechnung eines  $\alpha_1 \in (0, \alpha_0)$  kann aus den Informationen  $f(\mathbf{x}_{\text{opt}})$ ,  $\nabla f(\mathbf{x}_{\text{opt}})$  und  $f(\mathbf{x}_{\text{opt}} + \alpha_0 \mathbf{p})$  eine quadratische Funktion approximiert werden, aus deren Minimum sich  $\alpha_1$  ergibt. Genügt  $\alpha_1$  immer noch nicht den Bedingungen, kann mit der zusätzlichen Information  $f(\mathbf{x}_{\text{opt}} + \alpha_1 \mathbf{p})$  eine kubische Funktion als Approximation verwendet und ausgewertet werden, bis eine geeignete Schrittweite gefunden wird.

## 4.3 Statische Optimierung mit Nebenbedingungen

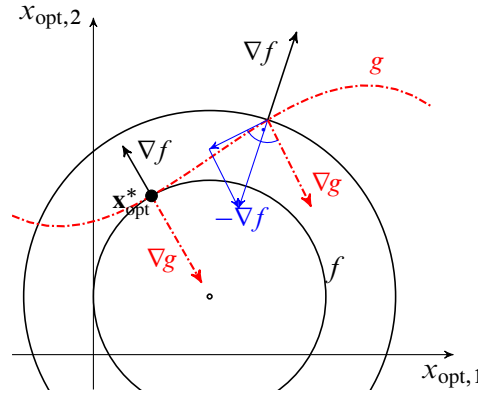
Nun werden auch die NB (4.5b) und (4.5c) in dem Problem (4.5a) beachtet. Zuerst werden nur GNB und anschließend beide Arten von NB untersucht.

### 4.3.1 Gleichungsnebenbedingungen

Das gleichungsbeschränkte Optimierungsproblem

$$\min f(\mathbf{x}_{\text{opt}}) \quad \text{u. B. v.} \quad \mathbf{g}(\mathbf{x}_{\text{opt}}) = \mathbf{0} \quad \text{mit} \quad \mathbf{g} : \mathbb{R}^{N_{\text{opt}}} \rightarrow \mathbb{R}^{m_g} \quad (4.22)$$

soll zuerst für den Fall  $N_{\text{opt}} = 2$  und einer einzelnen GNB ( $m_g = 1$ ) anhand Bild 4.2 geometrisch veranschaulicht werden. Betrachten wir einen beliebigen Punkt  $\mathbf{x}_{\text{opt}} \neq \mathbf{x}_{\text{opt}}^*$  auf  $g$ , so hat in diesem



**Bild 4.2:** Nur im Optimum sind  $\nabla f$  und  $\nabla g$  kollinear.

Punkt die Suchrichtung  $\mathbf{p}$ , die in Bild 4.2 einfach genau  $\mathbf{p} = -\nabla f$  gewählt wurde, eine Komponente parallel zu  $g$  oder anders ausgedrückt orthogonal zu  $\nabla g$ . In dieser Richtung kann also  $f$  unter Einhaltung von  $g(\mathbf{x}_{\text{opt}}) = 0$  reduziert werden. Im lokalen Optimum  $\mathbf{x}_{\text{opt}}^*$  sind  $\nabla f$  und  $\nabla g$  kollinear, sodass ein Skalar  $\lambda^*$  existiert, für den  $\lambda^* \nabla g(\mathbf{x}_{\text{opt}}^*) = \nabla f(\mathbf{x}_{\text{opt}}^*)$  ist.

Für jede GNB  $g_i$ ,  $i = 1, \dots, m_g$  gibt es einen sogenannten LAGRANGE-Multiplikator  $\lambda_i$ . Mit der LAGRANGE-Funktion [58, S. 310]

$$\mathcal{L}(\mathbf{x}_{\text{opt}}, \boldsymbol{\lambda}) = f(\mathbf{x}_{\text{opt}}) - \boldsymbol{\lambda}^\top \mathbf{g}(\mathbf{x}_{\text{opt}}) \quad (4.23)$$

lassen sich die *notwendigen Bedingungen 1. Ordnung* für ein lokales Minimum angeben:

$$\nabla_{\mathbf{x}_{\text{opt}}} \mathcal{L}(\mathbf{x}_{\text{opt}}^*, \boldsymbol{\lambda}^*) = \nabla f(\mathbf{x}_{\text{opt}}^*) - \mathbf{J}_g^\top(\mathbf{x}_{\text{opt}}^*) \boldsymbol{\lambda}^* = \mathbf{0} \quad (4.24a)$$

$$\nabla_{\boldsymbol{\lambda}} \mathcal{L}(\mathbf{x}_{\text{opt}}^*, \boldsymbol{\lambda}^*) = -\mathbf{g}(\mathbf{x}_{\text{opt}}^*) = \mathbf{0} \quad (4.24b)$$

Darin ist (man beachte die Definition des Gradienten als stehender Vektor auf S. 8)

$$\mathbf{J}_g^\top(\mathbf{x}_{\text{opt}}) = \nabla \mathbf{g}^\top(\mathbf{x}_{\text{opt}}) = [\nabla g_1(\mathbf{x}_{\text{opt}}), \dots, \nabla g_{m_g}(\mathbf{x}_{\text{opt}})] \quad (4.25)$$

die transponierte JACOBI-Matrix von  $\mathbf{g}(\mathbf{x}_{\text{opt}})$ . Die Bedingungen sind strukturell ähnlich zu (4.7) auf S. 50 und so können die Bestimmung der Suchrichtung und die Liniensuche prinzipiell wie in Abschnitt 4.2 erfolgen, nur wird die Gütefunktion  $f$  durch die LAGRANGE-Funktion  $\mathcal{L}$  ersetzt und  $\mathbf{x}_{\text{opt}}$  mit  $\boldsymbol{\lambda}$  erweitert. (4.7) impliziert dadurch sowohl (4.24a) als auch (4.24b). Jedoch steigt die Dimension des resultierenden Problems von  $N_{\text{opt}}$  auf  $N_{\text{opt}} + m_g$ .

Wir gehen im Folgenden davon aus, dass alle GNB linear unabhängig sind, sodass die JACOBI-Matrix vollen Rang hat. Diese Bedingung nennt man *linear independence constraint qualification* (LICQ) [58, S. 320]. Dann lassen sich alle zulässigen Suchrichtungen  $\mathbf{p}$  algebraisch und eindeutig bestimmen, was auch in Abschnitt 4.4.1 wichtig wird. Für alle zulässigen Suchrichtungen, bei denen aus den Bedingungen 1. Ordnung *nicht* hervorgeht, ob sie zu einer Ab- oder Zunahme der Kostenfunktion führen, fordert die *notwendige Bedingung 2. Ordnung*, dass zusätzlich

$$\mathbf{p}^\top \nabla_{\mathbf{xx}}^2 \mathcal{L}(\mathbf{x}_{\text{opt}}, \boldsymbol{\lambda}^*) \mathbf{p} \geq 0 \quad \forall \mathbf{p} \in \mathcal{C}_g(\mathbf{x}_{\text{opt}}^*) = \{ \mathbf{p} \in \mathbb{R}^{N_{\text{opt}}} \mid \mathbf{J}_g(\mathbf{x}_{\text{opt}}^*) \mathbf{p} = \mathbf{0} \} \quad (4.26)$$

erfüllt sein muss (vgl. [58, S. 330 ff.]). Darin bildet der Kern der JACOBI-Matrix die Menge aller zulässigen Suchrichtungen  $\mathcal{C}_g(\mathbf{x}_{\text{opt}}^*)$ .

## Einsetzverfahren

Existieren  $m_g$  unabhängige GNB in einem Optimierungsproblem mit  $N_{\text{opt}} > m_g$  Optimierungsvariablen, müssen  $m_g$  Variablen von den restlichen  $(N_{\text{opt}} - m_g)$  abhängig sein, da es nur noch  $(N_{\text{opt}} - m_g)$  Freiheitsgrade gibt. In manchen Fällen lassen sich die GNB analytisch in die Gütefunktion einsetzen, sodass anschließend ein Problem ohne GNB und nur noch mit  $(N_{\text{opt}} - m_g)$  Variablen zu lösen ist, wie es beim Aufstellen des QPs in Abschnitt 6.2.2 durchgeführt wird. Alternativ kann das Problem numerisch reduziert werden, wenn sich die entsprechende Teilmatrix der JACOBI-Matrix invertieren lässt (siehe hierfür „reduzierter Gradient“ in [60, S. 80 ff.]).

## Straffunktion

Die Straffunktion (engl. penalty function) dient dazu, eine GNB in die Gütefunktion einzubauen, sodass aus (4.22) das Ersatzproblem

$$\min_{\mathbf{x}_{\text{opt}}} f(\mathbf{x}_{\text{opt}}) + \frac{\mu}{2} \sum_{i=1}^{m_g} g_i^2(\mathbf{x}_{\text{opt}}) \quad (4.27)$$

wird, worin  $\mu > 0$  ein Gewichtungsparmeter (engl. penalty parameter) ist [58, S. 498 ff.]. Es kann sinnvoll sein,  $\mu$  in jeder Iteration zu erhöhen, um die GNB immer besser einzuhalten und das Originalproblem möglichst wenig zu verfälschen. Bild 4.3(a) zeigt den Verlauf der Straffunktion.

### 4.3.2 Ungleichungsnebenbedingungen

Zu den GNB werden zusätzlich die UNB (4.5c) hinzugenommen. Ähnlich wie bei GNB müssen  $\nabla f$  und die  $\nabla h_j$  aller aktiven UNB  $h_j$  im Optimum parallel sein, nur spielt das Vorzeichen des LAGRANGE-Multiplikators eine entscheidende Rolle. Die LAGRANGE-Funktion (4.23) wird daher unter Einführung des zusätzlichen LAGRANGE-Multiplikators  $\lambda_h \geq \mathbf{0}$  zu

$$\mathcal{L}(\mathbf{x}_{\text{opt}}, \lambda_g, \lambda_h) = f(\mathbf{x}_{\text{opt}}) - \lambda_g^\top \mathbf{g}(\mathbf{x}_{\text{opt}}) - \lambda_h^\top \mathbf{h}(\mathbf{x}_{\text{opt}}) \quad (4.28)$$

erweitert. Die *notwendigen Bedingungen 1. Ordnung* für ein lokales Minimum heißen **KARUSH-KUHN-TUCKER-Bedingungen** (*KKT-Bedingungen*) und lauten [58, S. 320 f.]:

$$\nabla_{\mathbf{x}_{\text{opt}}} \mathcal{L}(\mathbf{x}_{\text{opt}}^*, \lambda_g^*, \lambda_h^*) = \mathbf{0}, \quad (4.29a)$$

$$\mathbf{g}(\mathbf{x}_{\text{opt}}^*) = \mathbf{0}, \quad (4.29b)$$

$$\mathbf{h}(\mathbf{x}_{\text{opt}}^*) \geq \mathbf{0}, \quad (4.29c)$$

$$\lambda_h^* \geq \mathbf{0}, \quad (4.29d)$$

$$\lambda_h^{*\top} \mathbf{h}(\mathbf{x}_{\text{opt}}^*) = 0 \quad (4.29e)$$

Ist eine UNB  $h_j$  inaktiv, d. h.  $h_j(\mathbf{x}_{\text{opt}}^*) > 0$ , ist  $\lambda_{h,j}^* = 0$ , anderenfalls ist  $\lambda_{h,j}^* \geq 0$ . Diese *Komplementär-Bedingung* wird durch (4.29e) abgebildet. Der Gradient der LAGRANGE-Funktion nach

$\mathbf{x}_{\text{opt}}$  in (4.29a) ist auch in folgender Form darstellbar:

$$\begin{aligned}\nabla_{\mathbf{x}_{\text{opt}}} \mathcal{L}(\mathbf{x}_{\text{opt}}^*, \boldsymbol{\lambda}_g^*, \boldsymbol{\lambda}_h^*) &= \nabla f(\mathbf{x}_{\text{opt}}^*) - \mathbf{J}_g^\top(\mathbf{x}_{\text{opt}}) \boldsymbol{\lambda}_g^* - \mathbf{J}_h^\top(\mathbf{x}_{\text{opt}}) \boldsymbol{\lambda}_h^* \\ &= \nabla f(\mathbf{x}_{\text{opt}}^*) - \sum_{i \in \mathcal{E}} \lambda_{g,i}^* \nabla g_i(\mathbf{x}_{\text{opt}}^*) - \sum_{j \in \mathcal{A}} \lambda_{h,j}^* \nabla h_j(\mathbf{x}_{\text{opt}}^*)\end{aligned}\quad (4.30)$$

Die Indexmenge  $\mathcal{E}$  umfasst darin alle GNB und  $\mathcal{A}$  die *aktiven* UNB, sodass  $\forall j \notin \mathcal{A} : \lambda_{h,j}^* = 0$ . Für die *notwendige Bedingung 2. Ordnung* müssen nun alle aktiven NB beachtet werden, sodass

$$\begin{aligned}\mathcal{C}(\mathbf{x}_{\text{opt}}^*, \boldsymbol{\lambda}_h^*) &= \{ \mathbf{p} \in \mathbb{R}^{N_{\text{opt}}} \mid \nabla^\top g_i(\mathbf{x}_{\text{opt}}^*) \mathbf{p} = 0, \quad i \in \mathcal{E}, \\ &\quad \nabla^\top h_j(\mathbf{x}_{\text{opt}}^*) \mathbf{p} = 0, \quad j \in \mathcal{A} \text{ mit } \lambda_{h,j}^* > 0, \\ &\quad \nabla^\top h_j(\mathbf{x}_{\text{opt}}^*) \mathbf{p} \geq 0, \quad j \in \mathcal{A} \text{ mit } \lambda_{h,j}^* = 0 \}\end{aligned}\quad (4.31)$$

ist. Wiederum setzen wir voraus, dass die LICQ erfüllt ist, indem die Menge der Gradienten aller aktiven NB  $\{\nabla g_i, i \in \mathcal{E}\} \cup \{\nabla h_j(\mathbf{x}_{\text{opt}}^*), j \in \mathcal{A}\}$  linear unabhängig ist. Die KKT-Bedingungen lassen sich i. A. aufgrund der Komplementärbedingung (4.29e) nicht direkt lösen. Im Folgenden werden zwei ausgewählte Methoden zur numerischen Lösung angeführt.

### Sequentiell quadratische Programmierung

Durch eine TAYLOR-Reihenentwicklung von  $f$  um den aktuellen Punkt  $\mathbf{x}_{\text{opt}}^{(k)}$  herum und eine Linearisierung der NB  $\mathbf{g}$  und  $\mathbf{h}$  ergibt sich ein *quadratisches Problem* (QP) in der Form

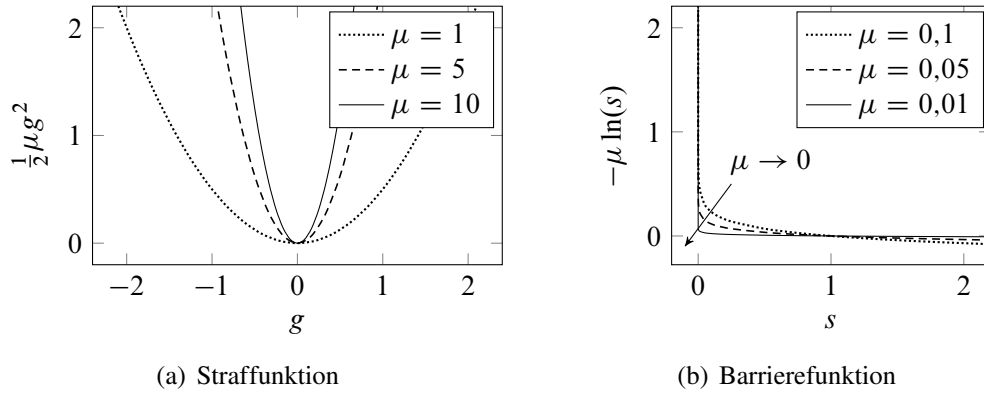
$$\min_{\mathbf{x}_{\text{opt}}} \quad \frac{1}{2} \mathbf{x}_{\text{opt}}^\top \mathbf{Q} \mathbf{x}_{\text{opt}} + \mathbf{c}^\top \mathbf{x}_{\text{opt}}, \quad \mathbf{Q} \succ 0 \quad (4.32a)$$

$$\text{u. B. v.} \quad \mathbf{A}_g \mathbf{x}_{\text{opt}} = \mathbf{b}_g, \quad (4.32b)$$

$$\mathbf{A}_h \mathbf{x}_{\text{opt}} \geq \mathbf{b}_h. \quad (4.32c)$$

Ein solches QP lässt sich allgemein effizient lösen. Zudem ist die hinreichende Bedingung 2. Ordnung aufgrund der Konvexität immer erfüllt, sodass jedes Optimum ein globales Optimum ist. In jeder Iteration wird nun die Reihenentwicklung an dem jeweils aktuellen Punkt erneut durchgeführt, weshalb man das Verfahren als *sequentiell quadratische Programmierung* (SQP) bezeichnet. Es ist ein weit verbreitetes Verfahren und gilt als eines der effizientesten Verfahren [58, S. 529]. Durch die Reihenentwicklung bzw. Linearisierung um den aktuellen Punkt hängt die Konvergenz stark von der Güte der Startlösung ab.

Für entsprechende Methoden sei auf die Literatur [58, Kap. 16] und [60, Kap. 5.4.3] verwiesen. Nur auf die größte Schwierigkeit beim Lösen eines QP, nämlich die UNB (4.32c), wird eingegangen: Wäre die Menge der aktiven NB im Optimum  $\mathcal{A}(\mathbf{x}_{\text{opt}}^*)$  zuvor bekannt, könnte man die LAGRANGE-Multiplikatoren so wählen, dass nur die aktiven NB berücksichtigt würden. Aber das ist dasselbe Problem, weshalb auch die KKT-Bedingungen nicht direkt gelöst werden können. Die oft verwendete *active set*-Methode bestimmt daher im aktuellen Iterationspunkt die Menge der aktiven UNB  $\mathcal{A}(\mathbf{x}_{\text{opt}}^{(i)})$  und berechnet nur unter Berücksichtigung dieser UNB als zusätzliche GNB einen Iterationsschritt. Der neue Iterationspunkt  $\mathbf{x}_{\text{opt}}^{(i+1)} = \mathbf{x}_{\text{opt}}^{(i)} + \mathbf{p}^{(i)}$  liegt zwangsläufig wieder auf



**Bild 4.3:** Während die Straffunktion für alle  $g \in \mathbb{R}$  definiert ist, müssen die Eingangswerte der Barrierefunktion strikt größer als null sein.

dem ursprünglichen *active set*. Ergibt  $\mathbf{x}_{\text{opt}}^{(i)} + \mathbf{p}^{(i)}$  einen unzulässigen Punkt, wird die Schrittweite reduziert und das *active set* um die entsprechende UNB ergänzt [60, S. 112 f.]. Ergibt sich mindestens ein negativer LAGRANGE-Multiplikator, wird die UNB mit dem kleinsten Multiplikator aus dem *active set* entfernt. Diese Strategie funktioniert in der Praxis meist gut, sie hat aber den Nachteil, dass die Auswahl von der Skalierung der UNB abhängt [58, S. 476]. Die im Minimum aktiven UNB werden also durch Ausprobieren verschiedener Mengen möglicher aktiver UNB bestimmt [60, S. 117], was bei einer hohen Anzahl von UNB zu einem beachtlichen Aufwand führt. Sowohl diese kombinatorische Komplexität als auch die mitunter schwierige Wahl eines zulässigen Startpunkts (engl. feasible point) kann mit der *Barriere-/Innere-Punkte-Methode* umgangen werden, bei der alle UNB in jedem Schritt per se erfüllt sind. Sie kann entweder zum Lösen von QPs oder als eigenständiges Verfahren zum Lösen nichtlinearer Probleme eingesetzt werden.

### Barriere-Methode/Innere-Punkte-Methode

Die Barriere-Methode wandelt eine UNB durch Einführung einer Schlupfvariablen  $s$  in eine GNB und eine Box-Restriktion  $s > 0$  um. Die Schlupfvariable taucht im Optimierungsproblem als zusätzliche Optimierungsvariable auf. Die Box-Restriktion kann mit geringem Mehraufwand berücksichtigt werden, indem die Schlupfvariable für  $s \rightarrow 0$  durch eine Barriere-Funktion im Gütemaß bestraft wird. Das Ersatzproblem für (4.5a) – (4.5c) lautet dann [58, S. 423 f.]:

$$\min_{\mathbf{x}_{\text{opt}}, \mathbf{s}} f(\mathbf{x}_{\text{opt}}) - \mu \sum_{i=1}^{m_h} \ln(s_i) \quad (4.33a)$$

$$\text{u. B. v. } \mathbf{g}(\mathbf{x}_{\text{opt}}) = \mathbf{0}, \quad (4.33b)$$

$$\mathbf{h}(\mathbf{x}_{\text{opt}}) - \mathbf{s} = \mathbf{0} \quad (4.33c)$$

Bild 4.3(b) zeigt, dass die Barriere nur Werte zulässt, die strikt im Inneren des zulässigen Bereichs liegen. Für  $\mu \rightarrow 0$  beeinflusst die Barrierefunktion das Originalproblem für  $s > 0$  nur marginal.

Frühe Barriere-Methoden nutzten keine Schlupfvariablen, sondern berechneten in der Gütefunktion direkt den Logarithmus von  $\mathbf{h}$ , um nur Werte für  $\mathbf{x}_{\text{opt}}$  zuzulassen, die die UNB erfüllen [58,

S. 583 ff.]. So entstand der Name *Innere-Punkte-Methode* (engl. *Interior-Point*, IP) als Synonym für die Barriere-Methode. Problematisch daran ist, dass auch die Startlösung die UNB erfüllen muss, da anderenfalls die Gütefunktion nicht ausgewertet werden kann.

Zusammen mit der SQP gehört das IP-Verfahren zu den mächtigsten Algorithmen zum Lösen hochdimensionaler Probleme [58, S. 563 f.]. Wie wir in Abschnitt 4.4 sehen werden, wurden inzwischen zahlreiche Methoden entwickelt, um trotz ungünstiger Startlösungen und vieler nichtlinearer UNB eine schnelle Konvergenz zu erzielen. Während SQP-Verfahren im Warmstart (Startlösung ist nahe am Optimum) sehr schnell konvergieren, fällt der Vorteil eines Warmstarts bei IP-Verfahren geringer aus, wenn der Barriereparameter  $\mu$  zu Beginn nicht optimal gewählt wird. Doch erweist sich eine perfekte Wahl von  $\mu$  in der Praxis als schwierig, zumal bei einer zu optimistischen Wahl die Gefahr besteht, dass die Konvergenz schlechter als beim Kaltstart ausfällt. Dafür bieten IP-Verfahren auch bei einer ungünstigen Startlösung oftmals eine sehr hohe Anfangskonvergenz. Auch weil das hier vorliegende Optimierungsproblem von vielen komplexen UNB dominiert wird, kommt ein IP-Verfahren zum Einsatz.

## 4.4 Nichtlineares Interior-Point-Verfahren Ipopt

In diesem Projekt wird der Interior-Point-Algorithmus IPOPT [86] verwendet. Im Folgenden werden die Grundlagen des IP-Verfahrens erläutert und ein paar wesentliche Aspekte aufgeführt, die den IPOPT-Algorithmus effizienter und robuster machen.

Die Barriere-Methode wandelt UNB mittels Schlupfvariablen in GNB um. Fasst man die Schlupfvariablen als zusätzliche primäre Variablen in  $\mathbf{x}_{\text{opt}}$  auf, existieren formal keine UNB nach (4.33c). Die Box-Restriktionen gemäß (4.6) auf S. 50 wollen wir separat behandeln und weiterhin mit Schlupfvariablen und Barriere-Funktion explizit abbilden. Wir werden im weiteren Verlauf sehen, dass sich diese Schlupfvariablen beim Aufstellen der KKT-Bedingungen eliminieren lassen, was sich prinzipiell auch auf nichtlineare UNB in (4.33c) übertragen lässt.  $\mathbf{x}_{\text{opt}}$  sei hier nur nach unten durch  $\mathbf{x}_{\text{opt}} \geq \mathbf{x}_{\text{lb}}$  begrenzt, da eine zusätzliche Obergrenze  $\mathbf{x}_{\text{opt}} \leq \mathbf{x}_{\text{ub}}$  zu strukturell identischen Gleichungen mit umgekehrten Vorzeichen führt. Wir betrachten also das Optimierungsproblem

$$\min_{\mathbf{x}_{\text{opt}}, \mathbf{s}} \quad f(\mathbf{x}_{\text{opt}}) - \mu \sum_{i=1}^{N_{\text{opt}}} \ln(s_i) \quad (4.34a)$$

$$\text{u. B. v.} \quad \mathbf{c}(\mathbf{x}_{\text{opt}}) = \mathbf{0}, \quad (4.34b)$$

$$\mathbf{x}_{\text{opt}} \geq \mathbf{x}_{\text{lb}} \quad \Leftrightarrow \quad \mathbf{x}_{\text{opt}} - \mathbf{x}_{\text{lb}} - \mathbf{s} = \mathbf{0}, \quad \mathbf{s} \geq \mathbf{0}. \quad (4.34c)$$

Mit der LAGRANGE-Funktion<sup>2</sup>

$$\mathcal{L}(\mathbf{x}_{\text{opt}}, \boldsymbol{\lambda}_c, \boldsymbol{\lambda}_{\text{lb}}, \mathbf{s}) = f(\mathbf{x}_{\text{opt}}) - \mu \sum_{i=1}^{N_{\text{opt}}} \ln(s_i) - \boldsymbol{\lambda}_c^\top \mathbf{c}(\mathbf{x}_{\text{opt}}) - \boldsymbol{\lambda}_{\text{lb}}^\top (\mathbf{x}_{\text{opt}} - \mathbf{x}_{\text{lb}} - \mathbf{s}) \quad (4.35)$$

<sup>2</sup>In der Literatur werden die LAGRANGE-Variablen bei der IP-Methode meist mit  $\mathbf{y}$  statt  $\boldsymbol{\lambda}_c$  und  $\mathbf{z}$  statt  $\boldsymbol{\lambda}_{\text{lb}}$  bezeichnet. Hier wird zugunsten der Konsistenz weiterhin  $\boldsymbol{\lambda}$  verwendet. Zudem wird auf das Superskript \* verzichtet.

folgen durch Differentiation nach  $(\mathbf{x}_{\text{opt}}, \boldsymbol{\lambda}_c, \boldsymbol{\lambda}_{\text{lb}}, \mathbf{s})$  die KKT-Bedingungen

$$\nabla f(\mathbf{x}_{\text{opt}}) - \mathbf{J}_c^\top(\mathbf{x}_{\text{opt}}) \boldsymbol{\lambda}_c - \mathbf{I} \boldsymbol{\lambda}_{\text{lb}} = \mathbf{0}, \quad (4.36a)$$

$$\mathbf{c}(\mathbf{x}_{\text{opt}}) = \mathbf{0}, \quad (4.36b)$$

$$\mathbf{x}_{\text{opt}} - \mathbf{x}_{\text{lb}} - \mathbf{s} = \mathbf{0} \quad \Leftrightarrow \quad \mathbf{S} = \mathbf{X}_{\text{opt}} - \mathbf{X}_{\text{lb}}, \quad (4.36c)$$

$$-\mu \mathbf{S}^{-1} \mathbf{i} + \boldsymbol{\lambda}_{\text{lb}} = \mathbf{0} \quad \Leftrightarrow \quad \mathbf{S} \boldsymbol{\lambda}_{\text{lb}} - \mu \mathbf{i} = \mathbf{0} \quad (4.36d)$$

mit den Diagonalmatrizen der jeweiligen Vektoren  $\mathbf{S} = \text{diag}(\mathbf{s})$  usw. Zudem ist  $\mathbf{J}_c$  die JACOBI-Matrix nach (4.25) auf S. 55 und  $\mathbf{i} = [1, \dots, 1]^\top$ . Die JACOBI-Matrix der Box-Restriktionen ist die Einheitsmatrix:  $\nabla^\top \mathbf{h}(\mathbf{x}_{\text{opt}}) = \nabla^\top(\mathbf{x}_{\text{opt}} - \mathbf{x}_{\text{lb}}) = \mathbf{I}$ . Die Barrierefunktion sichert implizit  $\mathbf{s} > \mathbf{0}$  und somit die Einhaltung der zu (4.29c) äquivalenten Forderung  $\mathbf{s} \geq \mathbf{0}$ . Die Umformung in (4.36d) ist eine Äquivalenzumformung, da  $\mathbf{S}$  nicht singulär ist. Sie zeigt, dass  $\mu > 0$  eine Relaxation der strengen Komplementärbedingung  $\boldsymbol{\lambda}_{\text{lb}}^\top \mathbf{s} = 0$  gemäß (4.29e) darstellt, indem sie für eine gegen null konvergierende Folge  $\{\mu_k\}$  lediglich eine paarweise Reduktion fordert.

Setzt man (4.36c) in (4.36d) ein, eliminiert man die Schlupfvariablen:

$$(\mathbf{X}_{\text{opt}} - \mathbf{X}_{\text{lb}}) \boldsymbol{\lambda}_{\text{lb}} - \mu \mathbf{i} = \mathbf{0}. \quad (4.37)$$

Wendet man das NEWTON-Verfahren aus (4.16) auf S. 52 auf (4.35) respektive auf das nichtlineare System (4.36) bzw. (4.37) in den primären und dualen Variablen  $(\mathbf{x}_{\text{opt}}, \boldsymbol{\lambda}_c, \boldsymbol{\lambda}_{\text{lb}})$  an, so erhält man das *primal-duale System* zur Bestimmung der Suchrichtungen  $\mathbf{p}_x$ ,  $\mathbf{p}_c$  und  $\mathbf{p}_{\text{lb}}$ :

$$\begin{bmatrix} \nabla_{\mathbf{x}\mathbf{x}}^2 \mathcal{L} & -\mathbf{J}_c^\top(\mathbf{x}_{\text{opt}}) & -\mathbf{I} \\ \mathbf{J}_c(\mathbf{x}_{\text{opt}}) & \mathbf{0} & \mathbf{0} \\ \boldsymbol{\Lambda}_b & \mathbf{0} & (\mathbf{X}_{\text{opt}} - \mathbf{X}_{\text{lb}}) \end{bmatrix} \begin{bmatrix} \mathbf{p}_x \\ \mathbf{p}_c \\ \mathbf{p}_{\text{lb}} \end{bmatrix} = - \begin{bmatrix} \nabla f(\mathbf{x}_{\text{opt}}) - \mathbf{J}_c^\top(\mathbf{x}_{\text{opt}}) \boldsymbol{\lambda}_c - \boldsymbol{\lambda}_{\text{lb}} \\ \mathbf{c}(\mathbf{x}_{\text{opt}}) \\ (\mathbf{X}_{\text{opt}} - \mathbf{X}_{\text{lb}}) \boldsymbol{\lambda}_{\text{lb}} - \mu \mathbf{i} \end{bmatrix} \quad (4.38)$$

Die Gl. (4.38) ist für  $\mathbf{x}_{\text{lb}} = \mathbf{0}$  sowie ein umgekehrt definiertes Vorzeichen von  $\boldsymbol{\lambda}_c$  identisch zu der entsprechenden Gleichung im Implementierungs-Paper von IPOPT, nämlich Gl. (9) in [86] und Gl. (2.5) in [57], auch wenn sich die obige Herleitung etwas unterscheidet.

Die letzte Zeile in (4.38) ergibt aufgelöst nach der Unbekannten, also der Suchrichtung

$$\mathbf{p}_{\text{lb}} = \mu \mathbf{X}_{\Delta}^{-1} \mathbf{i} - \boldsymbol{\lambda}_{\text{lb}} - \mathbf{X}_{\Delta}^{-1} \boldsymbol{\Lambda}_{\text{lb}} \mathbf{p}_x \quad \text{mit} \quad \mathbf{X}_{\Delta} := \mathbf{X}_{\text{opt}} - \mathbf{X}_{\text{lb}}. \quad (4.39)$$

Setzt man (4.39) in (4.38) ein, ergibt sich ein reduziertes, symmetrisches LGS, das zuerst gelöst wird [86]. Dabei sei im Moment  $\delta_w = 0$  und  $\delta_c = 0$ :

$$\begin{bmatrix} \nabla_{\mathbf{x}\mathbf{x}}^2 \mathcal{L} + \mathbf{X}_{\Delta}^{-1} \boldsymbol{\Lambda}_{\text{lb}} + \delta_w \mathbf{I} & \mathbf{J}_c^\top(\mathbf{x}_{\text{opt}}) \\ \mathbf{J}_c(\mathbf{x}_{\text{opt}}) & \delta_c \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{p}_x \\ -\mathbf{p}_c \end{bmatrix} = - \begin{bmatrix} \nabla \varphi(\mathbf{x}_{\text{opt}}) - \mathbf{J}_c^\top(\mathbf{x}_{\text{opt}}) \boldsymbol{\lambda}_c \\ \mathbf{c}(\mathbf{x}_{\text{opt}}) \end{bmatrix}, \quad (4.40)$$

$$\text{mit} \quad \varphi(\mathbf{x}_{\text{opt}}) := f(\mathbf{x}_{\text{opt}}) - \mu \sum_{i=1}^n \ln(\mathbf{x}_{\Delta,i}) \quad \Rightarrow \quad \nabla \varphi(\mathbf{x}_{\text{opt}}) = \nabla f(\mathbf{x}_{\text{opt}}) - \mu \mathbf{X}_{\Delta}^{-1} \mathbf{i}$$

Der Algorithmus terminiert, wenn der Optimalitätsfehler  $E_\mu(\mathbf{x}_{\text{opt}}, \boldsymbol{\lambda}_c, \boldsymbol{\lambda}_{\text{lb}})$  mit gewissen Skalierungsparametern  $s_d, s_c \geq 1$  eine vorgegebene Grenze  $\varepsilon_{\text{tol}} > 0$  unterschreitet [86]. Dabei ist

$$E_\mu(\mathbf{x}, \boldsymbol{\lambda}_c, \boldsymbol{\lambda}_{\text{lb}}) := \max \left\{ \frac{\|\nabla f(\mathbf{x}) - \mathbf{J}_c^\top(\mathbf{x}) \boldsymbol{\lambda}_c - \boldsymbol{\lambda}_{\text{lb}}\|_\infty}{s_d}, \|\mathbf{c}(\mathbf{x})\|_\infty, \frac{\|\mathbf{X} \boldsymbol{\lambda}_{\text{lb}} - \mu \mathbf{i}\|_\infty}{s_c} \right\}. \quad (4.41)$$



### 4.4.1 Inertia Correction und Feasibility Restoration Phase

Wenn die LICQ nicht erfüllt ist, also GNB linear abhängig sind, fällt der Rang der JACOBI-Matrix in der linken unteren Ecke von (4.40) ab, und das LGS kann aufgrund der Singularität nicht gelöst werden. Auf diese Bedingung sollte also schon bei der Formulierung des Optimierungsproblems geachtet werden. Kommt es trotzdem zum Rangabfall an dem aktuellen Wert von  $\mathbf{x}_{\text{opt}}$ , wird eine Korrektur („inertia correction“) vorgenommen, indem  $\delta_c > 0$  gewählt wird, wobei  $\delta_c \approx \sqrt{\varepsilon_{64}} \approx 10^{-8}$  ist (vgl. Maschinengenauigkeit  $\varepsilon_{64}$  in Anhang A) [86, S. 35].

Zudem folgt aus der notwendigen Bedingung 2. Ordnung (4.26) auf S. 55, dass die Projektion der Matrix in der oberen linken Ecke in den Kern der JACOBI-Matrix positiv definit sein muss. Diese Forderung ist erfüllt, wenn die Matrix genau  $n$  positive,  $m$  negative und keine Eigenwerte in null hat [86, S. 35]. Ist dies nicht der Fall, werden  $\delta_w, \delta_c \geq 0$  versuchsweise variiert, bis es erfüllt ist.

Ist die Matrix jedoch so schlecht konditioniert, dass dennoch keine Zerlegung möglich ist, wechselt IPOPT in die „Feasibility-Restoration-Phase“, in der versucht wird, unter temporärer Vernachlässigung des Gütemaßes einen Iterationspunkt zu finden, der die *Infeasibility*, also die Verletzung der NB reduziert [86, S. 38 ff.]. Hierfür wird ein Ersatzproblem gelöst, das zum einen die  $\ell_1$ -Norm der NB  $\|\mathbf{c}(\mathbf{x}_{\text{opt}})\|_1$  (Infeasibility) und zum anderen die quadratische Abweichung zum aktuellen Punkt  $\mathbf{x}_{\text{opt}}^{(k)}$  minimiert. Dieses neue Optimierungsproblem wird mit glatten Funktionen formuliert, um es mit der existierenden Barriere-Methode zu lösen. Im Erfolgsfall kann anschließend mit dem regulären Algorithmus fortgefahren werden; anderenfalls ist das Problem (*locally*) *infeasible*.

### 4.4.2 Filter Line-search und Second-Order-Correction

Nachdem die Suchrichtung aus (4.38) bestimmt wurde, muss über eine Liniensuche (vgl. Abschnitt 4.2.2) festgelegt werden, wie groß der Schritt sein soll. Denn weicht die Approximation (4.13) auf S. 51 vom tatsächlichen Verlauf ab, widerspricht ein voller NEWTON-Schritt ( $\alpha = 1$ ) womöglich der ARMIJO-Bedingung (4.19). Die Beurteilung eines Schritts ist mit NB schwieriger als ohne, da zwei Ziele vorliegen: Minimiere die Gütefunktion  $\varphi(\mathbf{x}_{\text{opt}})$  und die Verletzung der NB  $\vartheta(\mathbf{x}_{\text{opt}}) := \|\mathbf{c}(\mathbf{x}_{\text{opt}})\|_1$ . Ist ein Schritt inakzeptabel, wenn er ein Ziel verbessert, aber das andere verschlechtert? „Filter“ und „merit functions“ sollen die Akzeptanz eines Schritts beurteilen. Merit functions beurteilen eine Linearkombination aus dem Fortschritt in  $\varphi$  und  $\vartheta$ . Filter führen eine Liste mit Paaren  $(\varphi_k, \vartheta_k)$ , von denen kein Paar ein anderes dominiert, d. h. bei keinem anderen Paar  $(\varphi_i, \vartheta_i)$  ist sowohl  $\varphi_i \leq \varphi_k$  als auch  $\vartheta_i \leq \vartheta_k$  [28, S. 242]. Ein Schritt wird akzeptiert, wenn das resultierende Paar von keinem Paar im Filter dominiert wird. Dann wird das neue Paar ins Filter aufgenommen und das von ihm dominierte Paar gelöscht.

In IPOPT werden über die Suchrichtungen aus (4.40) und (4.39) sowie dem daraus folgenden Iterationspunkt die Schrittweiten  $\alpha_x^{(k)} \equiv \alpha_c^{(k)}, \alpha_{\text{lb}}^{(k)} \in (0, 1]$  abgeleitet, welche zuerst auf je eine maximale Schrittweite  $\hat{\alpha}_x^{(k)}, \hat{\alpha}_{\text{lb}}^{(k)}$  so begrenzt werden, dass die resultierenden Punkte  $\mathbf{x}_{\text{opt}}^{(k)} + \hat{\alpha}_x^{(k)} \mathbf{p}_x^{(k)}$  und  $\lambda_{\text{lb}}^{(k)} + \hat{\alpha}_{\text{lb}}^{(k)} \mathbf{p}_{\text{lb}}^{(k)}$  hinreichend weit von der Barriere entfernt sind [86, 83]. Es sind also unterschiedliche Schrittweiten für  $\lambda_{\text{lb}}$  und die anderen Variablen möglich. Dann folgt eine *Back-*

*tracking-Liniensuche*, in welcher für eine abnehmende Folge von Schrittweiten  $\hat{\alpha}_{x,l}^{(k)} = 2^{-l} \hat{\alpha}_x^{(k)}$ ,  $l = 0, 1, \dots$  die resultierenden Testpunkte von einem Filter auf Akzeptanz getestet werden [84, 86, 83]. Das Filter in IPOPT akzeptiert einen Testpunkt, wenn er *entweder* die Gütefunktion  $\varphi$  *oder* die Verletzung der NB  $\vartheta$  hinreichend reduziert. Da es vergangene Werte speichert, verhindert es einen Zyklus, in welchem abwechselnd Gütefunktion und NB verbessert werden. Folgt aus der Backtracking-Liniensuche eine inakzeptabel kleine Schrittweite, wechselt IPOPT in die bereits beschriebene *Feasibility-Restoration-Phase*.

Erfüllt ein Iterationspunkt die GNB bereits recht gut, kann es passieren, dass das Filter einen NEWTON-Schritt abweist, weil er sowohl die Gütefunktion als auch die GNB verschlechtert, obwohl er tatsächlich zu einem Punkt nahe des Optimums und somit zu einer guten Konvergenz führen würde (sogenannter MARATOS-Effekt [58, S. 440 f.]). Ein *Second-Order-Correction*-Schritt erzielt eine Reduzierung der Infeasibility, indem ein zusätzlicher NEWTON-Schritt für die NB am Punkt  $\mathbf{x}_{\text{opt}}^{(k)} + \hat{\alpha}_x^{(k)} \mathbf{p}_x^{(k)}$  durchgeführt wird [85, S. 35]. Es gibt verschiedene Umsetzungen, die u. a. in [28, 85, 86, 58] beschrieben sind.

### 4.4.3 Adaptive Barriere-Strategie

Die Wahl des Barriereparameters  $\mu$  kann auf verschiedene Arten erfolgen und hat großen Einfluss auf die Konvergenz des Algorithmus. Einige wesentliche Aspekte der in IPOPT implementierten Methoden werden im Folgenden kurz erklärt.

Der FIACCO-MCCORMICK-Algorithmus wählt einen konstanten Barriereparameter bis eine näherungsweise Lösung des Problems erreicht wird und reduziert ihn dann schrittweise [57]. Man kann für dieses *monotone* Verfahren zwar globale Konvergenz nachweisen, jedoch ist es manchmal ineffizienter als eine adaptive Strategie und kann eine gut gewählte Startlösung kaum ausnutzen, wenn nicht gleichzeitig der Anfangswert des Barriereparameters angepasst wird. Außerdem kann eine ungünstige Skalierung des Optimierungsproblems nicht ausgeglichen werden, wie man an (4.34a) sehen kann. Auch kommt es zu Problemen, wenn die Variablen die Grenzen des erlaubten Bereichs vorzeitig erreichen.

Eine *adaptive* Wahl von  $\mu$  erfolgt häufig über das *Dualitätsmaß*

$$\mu = \sigma_\mu \frac{\mathbf{x}_{\text{opt}}^\top \boldsymbol{\lambda}_{\text{lb}}}{N_{\text{opt}}} \quad (4.42)$$

mit dem „Centering-Parameter“  $\sigma_\mu > 0$  [57, 58, 60]. Zur Wahl von  $\sigma_\mu$  gibt es verschiedene Strategien, die oft heuristisch sind. Beispielsweise wird beim Solver LOQO das Dualitätsmaß durch die Abweichung des minimalen Komplementärwerts  $\xi$  zum Mittelwert dominiert [82, S. 243]:

$$\mu = 0,1 \left( \min \left\{ 0,05 \frac{1-\xi}{\xi}, 2 \right\} \right)^3 \frac{\mathbf{x}_{\text{opt}}^\top \boldsymbol{\lambda}_{\text{lb}}}{N_{\text{opt}}}, \quad \xi = \frac{\min_i x_{\text{opt},i} \lambda_{\text{lb},i}}{\mathbf{x}_{\text{opt}}^\top \boldsymbol{\lambda}_{\text{lb}} / N_{\text{opt}}} \quad (4.43)$$

Alternativ wird bei MEHROTRAS *Prädiktor-Korrektor-Methode* [52] zuerst (4.38) für  $\mu = 0$  gelöst („affine scaling step“), woraus sich mit (4.42) ein Zielwert für  $\mu$  ergibt. Anschließend folgen

ein primal-dualer Schritt und ein Korrekturschritt, siehe [52, 57]. [57] zeigt, dass MEHROTRAS Methode zwar für lineare und konvexe Probleme gut geeignet und bei einigen Problemen sehr effizient ist, jedoch nicht hinreichend robust im Nichtlinearen. Diese Aussage konnte bei Tests innerhalb dieser Arbeit bestätigt werden.

IPOPT bietet eine zusätzliche adaptive Strategie, die  $\mu$  durch eine approximative Minimierung einer „Quality-Function“ ermittelt. Dabei wird  $\mu$  gemäß (4.42) mit einem variablen Centering-Parameter  $\sigma_\mu$  gewählt und  $\mathbf{p}(\sigma_\mu)$  als die Lösung des Primal-dualen Systems (4.38) in Abhängigkeit von  $\sigma_\mu$  definiert, sodass sich eine Schrittweite  $\hat{\alpha}(\sigma_\mu)$  (vgl. Abschnitt 4.4.2) abhängig von  $\sigma_\mu$  ergibt [57, S. 7]. Ziel ist es,  $\sigma_\mu$  so zu wählen, dass die KKT-Bedingungen bestmöglich eingehalten werden. Dafür könnte ein Term ähnlich dem Optimalitätsfehler (4.41) mit einer  $\ell_2$ -Norm abhängig von  $\sigma_\mu$  minimiert werden, was jedoch zu einem aufwendigen nichtlinearen Problem führen würde. Daher minimiert IPOPT die weitaus weniger aufwendige teils lineare Funktion [57]

$$\begin{aligned} q_L(\sigma_\mu) = & (1 - \hat{\alpha}_{lb}(\sigma_\mu))^2 \|\nabla f(\mathbf{x}_{opt}) - \mathbf{J}_c^T(\mathbf{x}_{opt}) \boldsymbol{\lambda}_c - \boldsymbol{\lambda}_{lb}\|_2^2 + \dots \\ & + (1 - \hat{\alpha}_x(\sigma_\mu))^2 \|\mathbf{c}(\mathbf{x}_{opt})\|_2^2 + \dots \\ & + \|(\mathbf{X}_{opt} + \hat{\alpha}_x(\sigma_\mu) \mathbf{p}_x(\sigma_\mu)) (\boldsymbol{\Lambda}_{lb} + \hat{\alpha}_{lb}(\sigma_\mu) \mathbf{p}_{lb}(\sigma_\mu)) \mathbf{i}\|_2^2. \end{aligned} \quad (4.44)$$

Die Vereinfachung besteht hauptsächlich darin, dass die Ausdrücke innerhalb der Norm in den ersten beiden Termen nicht mehr von  $\sigma_\mu$  abhängen, also ihr Verlauf linearisiert wurde. Der rechenintensivste Teil ist die Berechnung der maximalen Schrittweiten  $\hat{\alpha}_x$  und  $\hat{\alpha}_{lb}$  sowie der letzte Ausdruck in (4.44). Ein großer Vorteil dieser Methode ist, dass sie invariant gegenüber einer Skalierung der Gütefunktion ist, und so von den Vorteilen des (skalierungsinvarianten) NEWTON-Verfahrens profitieren kann. Daher wurde auch in den numerischen Tests in [57, S. 10 f.] auf eine Skalierung verzichtet.

Da sich der Barriereparameter bei dieser adaptiven Methode in jeder Iteration ändern kann, kann keine globale Konvergenz der Barriere-Strategie nachgewiesen werden, wie es bei der FIACCO-MCCORMICK-Methode mit einer entsprechenden Liniensuche und einer Merit-Function oder einem Filter der Fall ist [57, S. 8]. Daher überwacht IPOPT den Fortschritt des primal-dualen Schritts, z. B. in Form des KKT-Fehlers des nichtlinearen Originalproblems. Solange der Fortschritt hinreichend groß ist, kann der Barriereparameter im sogenannten „free mode“ gemäß (4.44) frei gewählt werden. Ist kein hinreichender Fortschritt gegeben, kehrt der Algorithmus zur FIACCO-MCCORMICK-Strategie mit der Filter-Liniensuche („monotone mode“) zurück, bis der Fortschritt wieder so groß ist, dass in den Free-Mode gewechselt werden kann. Diese Strategie wird „globalization method“ genannt [57].

## 5 Optimierungsproblem

Aus dem zeitoptimalen Gütemaß (3.2) sowie dem Fahrzeugmodell (3.10) und einigen weiteren Nebenbedingungen lässt sich ein Optimierungsproblem aufstellen. In Kapitel 4 wurden die mathematischen Schritte, welche zum Lösen eines Optimierungsproblems mit den hier eingesetzten Verfahren notwendig sind, allgemein erklärt. Nun muss das Optimierungsproblem aus Kapitel 3 so formuliert werden, dass die Methoden aus Kapitel 4 zur Lösung dieses Problems verwendet werden können. Außerdem wird die mathematische Formulierung einiger Nebenbedingungen so angepasst, dass sie besonders effizient berücksichtigt werden können. Dies ist gerade in Hinblick auf eine echtzeitfähige Trajektorienplanung unabdingbar.

Während der Fokus in Kapitel 4 also auf den mathematischen Lösungsmethoden für ein bestehendes Problem lag, beschreibt dieses Kapitel das zu lösende Problem selbst, ohne nochmals auf den Lösungsweg einzugehen. Lediglich in Abschnitt 5.1, wo aus dem dynamischen (kontinuierlichen) Problem in Kapitel 3 ein statisches (diskretes) Optimierungsproblem erstellt wird, müssen Details der Diskretisierung erneut aufgegriffen werden, weil die hier verwendete Methode abweichend ist und sich daraus wichtige Aspekte der anschließend beschriebenen Modellierung und Formulierung der Aktordynamik ergeben.

Das Optimierungsproblem muss im IPOPT-Interface in der Form [87]:

$$\min_{\mathbf{x}_{\text{opt}}} f(\mathbf{x}_{\text{opt}}) \quad (5.1a)$$

$$\text{u. B. v. } \mathbf{x}_{\text{lb}} \leq \mathbf{x}_{\text{opt}} \leq \mathbf{x}_{\text{ub}} \quad (5.1b)$$

$$\mathbf{c}_{\text{lb}} \leq \mathbf{c}(\mathbf{x}_{\text{opt}}) \leq \mathbf{c}_{\text{ub}} \quad (5.1c)$$

implementiert sein, wobei GNB durch gleiche untere und obere Grenzen  $c_{\text{lb}} = c_{\text{ub}}$  ausgedrückt werden. In Abschnitt 5.2 wird die Gütefunktion (5.1a) aufgestellt, welche aus der Zeitoptimalitätsbedingung und weiteren Güte-termen besteht. Anschließend werden die Nebenbedingungen (5.1b) und (5.1c) in Abschnitt 5.3 beschrieben.

Zudem müssen in jeder Iteration die Ableitungen der LAGRANGE-Funktion berechnet werden, also der Gradient  $\nabla f(\mathbf{x}_{\text{opt}})$  und die JACOBI-Matrix  $\nabla^T \mathbf{c}(\mathbf{x}_{\text{opt}})$ . Wie in Kapitel 4 erklärt, benötigt IPOPT für das NEWTON-Verfahren auch die zweite Ableitung der LAGRANGE-Funktion, also

$$\frac{\partial^2}{\partial \mathbf{x}_{\text{opt}}^2} \mathcal{L}(\mathbf{x}_{\text{opt}}, \lambda_{\text{obj}}, \boldsymbol{\lambda}_c) = \lambda_{\text{obj}} \nabla^2 f(\mathbf{x}_{\text{opt}}) + \sum_i \lambda_{c,i} \nabla^2 c_i(\mathbf{x}_{\text{opt}}) \quad (5.2)$$

wobei die HESSE-Matrizen der Gütefunktion sowie aller NB mit ihren jeweiligen LAGRANGE-Multiplikatoren  $\lambda_{c,i}$  gewichtet und aufsummiert werden. Die Ableitungen können zwar approximiert werden, jedoch werden sie in der vorliegenden Arbeit exakt, also analytisch, berechnet. Hier auf sowie auf ihre Implementierung geht Abschnitt 5.4 ein. Das Kapitel schließt in Abschnitt 5.5 mit Simulationsergebnissen der Trajektorienplanung.

## 5.1 Diskretisierung und Aktordynamik

Zur Formulierung des statischen Optimierungsproblems aus den dynamischen Gleichungen (3.2) auf S. 25 und (3.10) auf S. 28 wird das Optimierungsproblem diskretisiert. Dieser Vorgang, den man auch als *Transkription* bezeichnet, wurde in Abschnitt 4.1 beschrieben. Dort wurden speziell die zwei Möglichkeiten Teil- und Volldiskretisierung unterschieden und ihre Vor- und Nachteile erklärt. In der vorliegenden Arbeit wird die Volldiskretisierung bezüglich der Stellgrößen so modifiziert, dass sie die Vorteile beider Methoden kombiniert und so die Rechenzeit signifikant reduziert, was in Abschnitt 5.1.1 erklärt wird.

Kurz zusammengefasst aus Abschnitt 4.1 wird bei der Teildiskretisierung lediglich der Stellgrößenverlauf diskretisiert, sodass der Optimierungsvektor  $\mathbf{x}_{\text{opt}}$  nur aus Stellgrößen  $\mathbf{u}$  besteht – hier also aus  $F_M$ ,  $F_B$  und  $\delta$ . Die Zustände  $\mathbf{x}$  ergeben sich in jeder Iteration durch eine separat ausgeführte Simulation der System-DGLs. Bei der Volldiskretisierung wird zusätzlich die Trajektorie an denselben Stellen diskretisiert und die System-DGLs werden in Form von GNB im Optimierungsproblem berücksichtigt. Der Vorteil der Teildiskretisierung besteht in der minimalen Anzahl an Optimierungsvariablen, die gerade der Anzahl der Freiheitsgrade entspricht. Ein großer Nachteil ist, dass alle Terme nach den Optimierungsvariablen, also den Stellgrößen, differenziert werden müssen, was bei komplexen Zusammenhängen einen immensen Mehraufwand darstellt:

Selbst eine scheinbar einfache UNB, wie die Begrenzung von  $d(s)$  auf die Streckenränder über  $d_{\text{max, re}}(s) \leq d(s) \leq d_{\text{max, li}}(s)$ , stellt ein kaum lösbares Problem dar, wenn  $d(s)$  unter Berücksichtigung des Einspurmodells je zweimal nach  $F_M$ ,  $F_B$  und  $\delta$  differenziert werden soll. Ist  $d(s)$  aber eine Optimierungsvariable, so ist die UNB keine nichtlineare Gleichung nach (5.1c), sondern eine simple Box-Restriktion nach (5.1b). CASANOVA verwendet dennoch eine Teildiskretisierung und thematisiert diese Problematik so: „The mathematical description of these constraints is crucial and may affect the computational efficiency of the optimisation procedure.“ [20, S. 106 ff.] Er setzt deswegen an Stellen, wo der Fzg-Schwerpunkt nahe am Rand ist, lokal einen quadratischen Strafterm ein, der den SP in Richtung Fahrbahnmitte drückt. Durch die Segmentierung der Strecke sollen die Stellgrößen jeweils nur einen kurzen Streckenabschnitt beeinflussen.

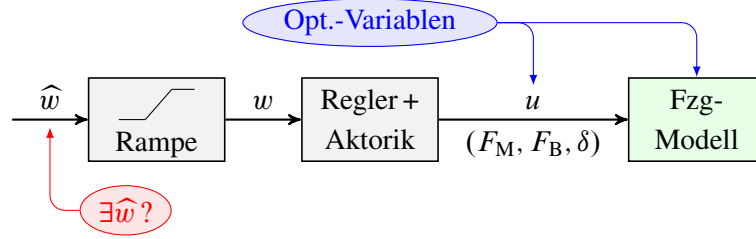
In der vorliegenden Arbeit wird aufgrund der genannten Problematik zunächst eine Volldiskretisierung durchgeführt, wodurch sich bei einer Strecke mit  $(N + 1)$  Abtastpunkten  $s_1, \dots, s_{N+1}$  folgender Vektor der Optimierungsvariablen ergibt:

$$\mathbf{x}_{\text{opt}} = \begin{cases} \begin{bmatrix} \mathbf{u}_M^\top & \mathbf{u}_B^\top & \delta^\top & \mathbf{v}^\top & \mathbf{d}^\top & \boldsymbol{\beta}^\top & \boldsymbol{\psi}_t^\top & \dot{\boldsymbol{\psi}}^\top \end{bmatrix}^\top \in \mathbb{R}^{8N}, & \text{Rundkurs} \\ \begin{bmatrix} \mathbf{u}_M^\top & \mathbf{u}_B^\top & \delta^\top & \mathbf{v}_{++}^\top & \mathbf{d}_{++}^\top & \boldsymbol{\beta}_{++}^\top & \boldsymbol{\psi}_{t++}^\top & \dot{\boldsymbol{\psi}}_{++}^\top \end{bmatrix}^\top \in \mathbb{R}^{8N+5}, & \text{offene Strecke} \end{cases}$$

$$\text{mit } \mathbf{u}_M^\top = [u_{M,1} \ \dots \ u_{M,N}]^\top, \quad \mathbf{u}_B^\top = [u_{B,1} \ \dots \ u_{B,N}]^\top \quad \text{usw.,}$$

$$\text{bzw. } \mathbf{v}_{++}^\top = [v_1 \ \dots \ v_N \ v_{N+1}]^\top \quad \text{usw.,}$$

$$\text{wobei } \mathbf{u}_M = 10^{-3} \mathbf{F}_M \quad \text{und} \quad \mathbf{u}_B = 10^{-3} \mathbf{F}_B \quad \text{skaliert werden.}$$



**Bild 5.1:** Existiert eine Steuerfolge  $\hat{w}$ , welche zu dem gewünschten Verlauf der Optimierungsvariable  $u$  führt? [95]

Die recht ungewöhnliche Sortierung der Variablen nach Zuständen anstelle von Abtastschritten hat den Vorteil, dass der komplette Verlauf eines einzelnen Zustands zur Programmlaufzeit sehr effizient aus dem Optimierungsvektor  $\mathbf{x}_{\text{opt}}$  extrahiert werden kann, weil alle benötigten Variablen im Speicher direkt hintereinander liegen.

Zur Diskretisierung der DGL (3.35) auf S. 36 wird nun ein explizites EULER-Verfahren verwendet, weil dies zu einer besonders dünn besetzten HESSE-Matrix führt, wie es später in (5.38) zu sehen sein wird. Somit ergibt sich mit dem diskreten Zustands- bzw. Eingangsvektor

$$\mathbf{x}_k = \begin{bmatrix} v_k & d_k & \beta_k & \psi_{t,k} & \dot{\psi}_k \end{bmatrix}^T, \quad \mathbf{u}_k = \begin{bmatrix} F_{M,k} & F_{B,k} & \delta_k \end{bmatrix}^T \quad (5.3)$$

und den DGLs (3.34a), (3.7), (3.34b), (3.9) und (3.32c) respektive (3.10) die vektorielle Differenzengleichung

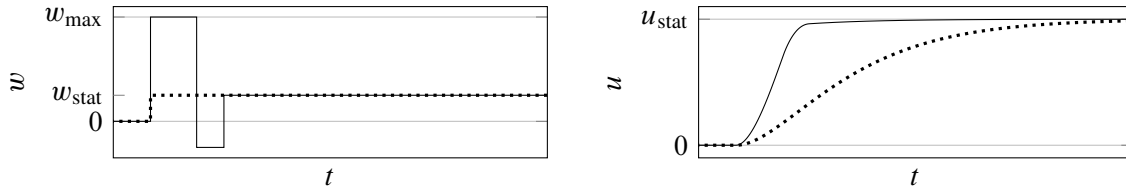
$$\mathbf{x}_{k+1} = \mathbf{x}_k + h_k \cdot \sigma_k(\mathbf{x}_k) \cdot \mathbf{f}_{\text{mod},k}(\mathbf{x}_k, \mathbf{u}_k), \quad h_k := s_{k+1} - s_k, \quad k = 1, \dots, N. \quad (5.4)$$

Für einen Rundkurs sind Anfangs- und Endpunkt der Referenz identisch,  $P_{\text{ref}}(s_{N+1}) \equiv P_{\text{ref}}(s_1)$ . Daher muss dort auch zwangsläufig der Zustandsvektor  $\mathbf{x}_{N+1} \equiv \mathbf{x}_1$  sein, weshalb  $\mathbf{x}_{N+1}$  bei Rundkursen nicht als separate Optimierungsvariable mitgeführt wird.

### 5.1.1 Modellierung der Aktordynamik

Bisher wurde stillschweigend davon ausgegangen, dass man für die Eingangsgrößen des Modells  $\mathbf{u}_k$  einen nahezu beliebigen Verlauf vorgeben könne. Jedoch muss die Dynamik der beteiligten Aktoren (Motor, Bremse, Lenkung) mit ihren entsprechenden unterlagerten Regelkreisen beachtet werden, wie es in Bild 5.1 verdeutlicht wird. Der klassische Weg der Volldiskretisierung würde diese Dynamik ins Modell aufnehmen. Oft wird sie durch ein PT<sub>1</sub>- oder PT<sub>2</sub>-Glied approximiert, wobei der Eingang aus Sicherheits- oder Komfortgründen auf eine Maximalrate begrenzt ist, was in Bild 5.1 durch die Rampe angedeutet ist. Um das Problem zu vereinfachen, könnte die Stellrate  $|\dot{u}|$  wie in [96, 90] durch lineare UNB begrenzt werden. Dieser Ansatz scheidet hier aber aus, weil weder ein glatter Verlauf von  $w$  erzielt noch ein PT<sub>2</sub>-Verhalten nachgebildet werden kann.

Modelliert man die Dynamik, ergeben sich bereits für ein PT<sub>1</sub>-Glied im Optimierungsproblem bei  $N$  DP weitere  $N$  Optimierungsvariablen pro Aktor für  $w$  sowie  $N$  zusätzliche GNB für jeden Aktor, wodurch das Optimierungsproblem extrem wächst. Möchte man auch noch die Stellrate am



**Bild 5.2:** Ein Überschwingen in  $w$  führt zu einem schnelleren Anstieg in  $u$ .

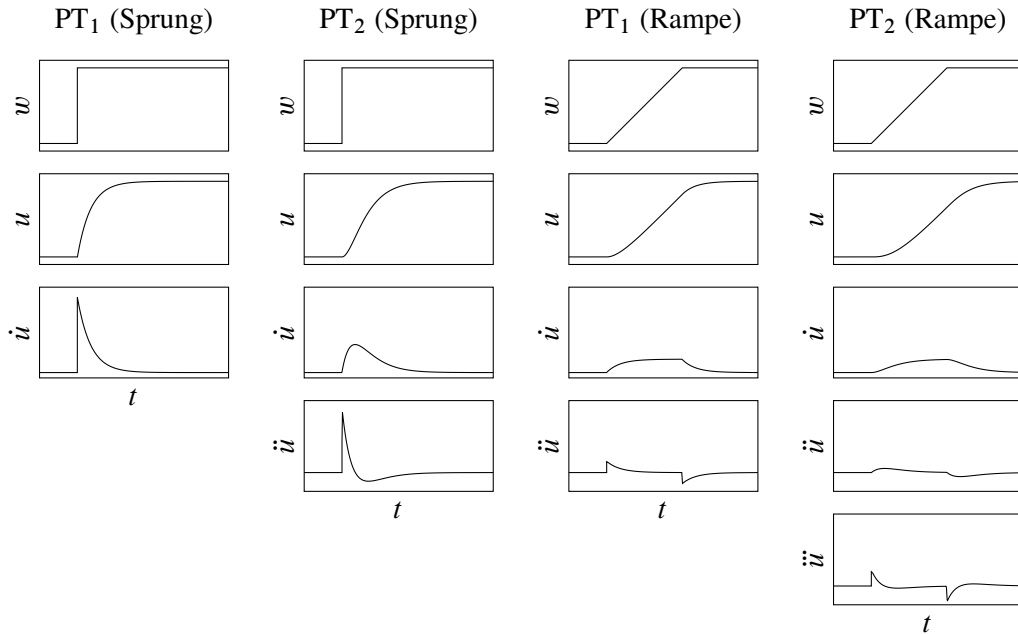
Eingang begrenzen, ergeben sich weitere  $2N$  UNB. Diese Stellratenbegrenzung, also die Rampe in Bild 5.1, ist hierbei sehr wichtig und es genügt für ein natürliches Fahrverhalten nicht, die Stellgröße  $w$  über eine Box-Restriktion auf einen Maximalwert zu begrenzen, was am folgenden Beispiel veranschaulicht werden soll: Gehen wir davon aus, dass der geschlossene Regelkreis der Lenkaktorik ein  $PT_2$ -Verhalten habe. Dann folgt auf einen Sprung im Lenkrad-Moment ein einmal stetig differenzierbarer Verlauf des Lenkwinkels (= *Rad*-Lenkwinkel) mit der entsprechenden Zeitkonstante, wie in Bild 5.2 gepunktet dargestellt ist. In vielen Situationen würde sich die Fahrzeit reduzieren, wenn die Zeitkonstante kürzer wäre. Und die Optimierung ist nun einmal ebenso eine *Optimalsteuerung*. Daher wird sie für das Modell eine derartige Vorsteuerung berechnen, dass sich der Lenkwinkel schneller ändern wird. Das erreicht sie, indem sie  $w$  sprunghaft (die Stellrate  $|\Delta w|$  sei ja nicht begrenzt) bis in die Stellgrößenbegrenzung  $|w| \leq w_{\max}$  treibt und dann sukzessive dem Zielwert annähert, was bei diskreten Systemen im Extremfall zum *Deadbeat*-Verhalten führt. Das Prinzip zeigt Bild 5.2 exemplarisch. Diese Steuerung mag zwar zeitoptimal und somit auch das mathematische Optimum sein, ist aber dennoch keine sinnvolle Lösung, weil sie sich im Fzg recht unangenehm anfühlt, da der Verlauf von  $u$  weniger glatt ist. Also ist eine Stellratenbegrenzung von  $w$  wichtig. Doch auf welchen Wert soll sie begrenzt werden? – Eine gute allgemeingültige Grenze, die langsam genug ist, um ein angenehmes Fahrverhalten zu erzielen, und schnell genug, um auch in kritischen Situationen Lösungen zu finden und nicht unnötig einzuschränken, lässt sich schwer angeben. Deswegen wird im Folgenden ein Ansatz vorgestellt, der diesen Widerspruch lösen kann und sich zudem auch sehr effizient im Optimierungsproblem einbauen lässt.

Eine markante Eigenschaft eines  $PT_n$ -Glieds ist, dass die  $n$ -te Ableitung seiner Sprungantwort unstetig ist. Eine Rampe als Eingangsgröße erhöht die Differenzierbarkeit der Systemantwort dann um eins. Bild 5.3 zeigt eine Übersicht über den qualitativen Verlauf der Antwort eines  $PT_1$ - und  $PT_2$ -Glieds auf sprung- und rampenförmige Eingangsgrößen und visualisiert die Differenzierbarkeit des Ausgangssignals. Dieser Zusammenhang wird nun genutzt, um am Beispiel eines  $PT_1$ -Glieds aus dem Verlauf von  $w$  eine Bedingung für  $u$  abzuleiten. Diskretisiert man die Übertragungsfunktion eines  $PT_1$ -Glieds mit der Zeitkonstante  $\tau$

$$G(s) = \frac{U(s)}{W(s)} = \frac{1}{\tau s + 1} \quad (5.5)$$

mit einem Halteglied 0. Ordnung, ergibt sich bei der Abtastzeit  $T_s$  die Differenzengleichung

$$w_k = \frac{1}{1-c} u_{k+1} - \frac{c}{1-c} u_k, \quad c = e^{-T_s/\tau}. \quad (5.6)$$



**Bild 5.3:** Typischer Verlauf des Ausgangssignals  $u$  eines  $PT_1$ - bzw.  $PT_2$ -Glieds (mit reellen Polstellen) bei einer sprung- bzw. rampenförmigen Eingangsgröße  $w$ . Zusätzlich sind diejenigen Ableitungen von  $u$  dargestellt, die endlich sind.

Mit einer zusätzlichen Stellratenbegrenzung

$$|\Delta w| = |w_{k+1} - w_k| \leq \Delta w_{\max} \quad (5.7)$$

ergibt sich eine Differenzengleichung 2. Ordnung der Form

$$|\alpha u_k + \beta u_{k+1} + \gamma u_{k+2}| \leq \Delta w_{\max}, \quad (5.8)$$

mit zuerst noch unbestimmten Koeffizienten  $\alpha, \gamma > 0$  und  $\beta < 0$ . Man erhält so eine UNB, die ausschließlich von der Variable  $u$  abhängt, welche sowieso als Optimierungsvariable existiert. Dieser Ansatz geht also gerade den umgekehrten Weg zur Teildiskretisierung, indem die Zustände als Optimierungsvariablen verwendet werden und die Stellgrößen implizit aus den Zuständen abgeleitet werden. So kombiniert man zwei Vorteile, nämlich zum einen alle Zustände im Optimierungsproblem verwenden zu können, und zum anderen die Problemdimension zu reduzieren.

Die Koeffizienten  $\alpha, \beta, \gamma$  in (5.8) hängen bei einem  $PT_n$ -Glieder gemäß (5.6) über e-Funktionen von der Abtastzeit  $T_s$  ab, welche wiederum von  $\sigma$  abhängt. Da diese Abhängigkeit komplex ist, sollte untersucht werden, ob sie in dieser Form notwendig ist: Das  $PT_n$ -Verhalten zur Modellierung der unterlagerten Regelkreise wird meist angesetzt, damit sich eine lineare DGL ergibt, was in einem nichtlinearen Optimierungsproblem jedoch keinen Vorteil bietet. Ein  $PT_n$ -Glieder zeichnet sich durch eine fixe Zeitkonstante aus, die nicht von der Sprunghöhe abhängt. Genau dieser Zusammenhang liegt in der Realität aber keineswegs vor: Wenn der Motor „auf Drehzahl ist“ wird er durch den Turbo schneller ansprechen als aus dem Stand, und auch die Bremse wird schneller ihre maximale Wirkung entfalten, wenn die Bremsbacken schon an der Bremscheibe anliegen. Es gibt also keinen Grund, an der Approximation durch ein  $PT_n$ -Glieder festzuhalten.



Nimmt man statt des  $PT_1$ -Glieds (5.5) ein integrierendes Verhalten an, bei dem die Sprungantwort rampenförmig bis zum Endwert läuft, ist die Zeit bis zum Erreichen des Endwerts proportional zur Sprunghöhe. Dann ergibt sich für  $\alpha$ ,  $\beta$  und  $\gamma$  der Zusammenhang  $\beta = -2\alpha$  und  $\gamma = \alpha$ , sodass die Differenzengleichung (5.8) dem klassischen *Differenzenquotienten 2. Ordnung* entspricht. Das heißt für ein System  $n$ -ter Ordnung wird der Betrag der  $(n+1)$ -ten Ableitung des Ausgangs  $u$  auf einen endlichen festen Wert begrenzt. Diese UNB für das I-Verhalten ist gerade äquivalent zu der zuvor angeführten notwendigen Bedingung, dass für ein  $PT_n$ -Glied mit Stellratenbegrenzung die  $(n+1)$ -te Ableitung des Ausgangs endlich sein muss, wie es in Bild 5.3 dargestellt ist.

### 5.1.2 Formulierung der Aktordynamik als quadratische Straffunktion

Nun gehen wir noch einen Schritt weiter und ersetzen die UNB (5.8) durch eine quadratische Straffunktion:

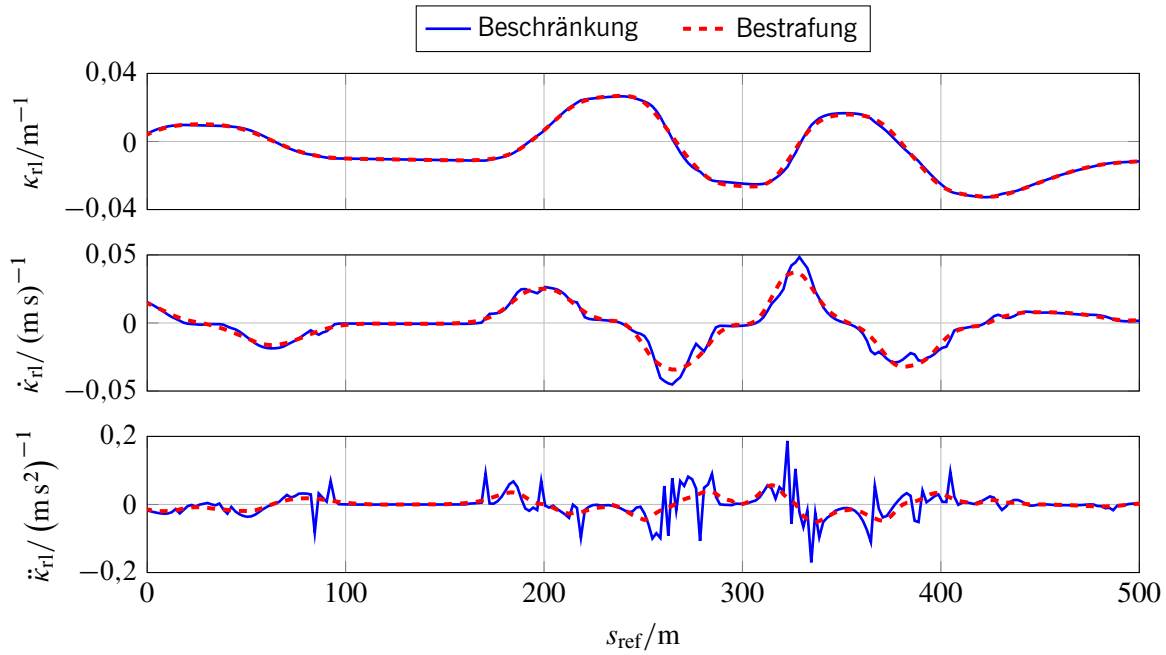
$$\min \quad \Delta w^2 = (\alpha u_k + \beta u_{k+1} + \gamma u_{k+2})^2 \quad (5.9)$$

So entfallen zusätzliche UNB, was (testweise) ca. 10 % Rechenzeit einspart. Zudem reduziert dieses zusätzliche Gütemaß die Dynamik in der Lösung und führt zu spürbar erhöhtem Fahrkomfort. Trotzdem lässt der Güte-term in einer Notsituation auch eine schnelle Änderung der Stellgröße zu, sodass man die zuvor beschriebene Problematik umgeht, für jene widersprüchlichen Anforderungen an eine Obergrenze der Stellrate einen allgemeingültigen Kompromiss finden zu müssen. Denn so wird die Obergrenze im wahrsten Sinne des Wortes optimiert.

Es kann durch die Straffunktion zwar nicht sicher gestellt werden, dass die Grenzen der Aktordynamik stets eingehalten werden, jedoch haben Tests gezeigt, dass der Trajektorienfolgeregler, welchem ja nur die Zustände und nicht die Eingänge übergeben werden, dem glatteren Verlauf auch in hochdynamischen Situationen besser folgen kann. Man darf das Gütemaß (5.9) nur nicht zu stark gewichten, da dies zu Überspringen führen würde, was einem Passagier unmotiviert erscheint. Das Überspringen ist prinzipiell vergleichbar mit der Oszillation einer polynomialen Interpolation, welche entstehen kann, wenn man die Polynomordnung erhöht, um einen glatteren, d. h. häufiger stetig differenzierbaren, Verlauf zu erhalten. Diese Problematik wird später anhand von Bild 5.7 auf S. 78 nochmals thematisiert.

Bild 5.4 zeigt die Krümmung einer optimierten Trajektorie für einen Streckenabschnitt auf einem Handlings-Kurs für die Beschränkung nach (5.8) im Vergleich zur Bestrafung nach (5.9). Gerade an den Ableitungen sieht man deutlich, wie viel glatter der Verlauf wird. Dabei erhöht sich die simulierte Rundenzeit nur marginal um  $\Delta T_{\text{lap}} < 1 \%$ .

Darüber hinaus hat die Straffunktion (5.9) einen weiteren Vorteil gegenüber der UNB (5.8). Um bei Verwendung der UNB einen rampenförmigen Verlauf in  $w$  zu erhalten, muss gemäß Bild 5.3 bei einem Aktormodell  $n$ -ter Ordnung die  $(n+1)$ -te Ableitung von  $u$  endlich sein, also beschränkt werden. Man kann in Bild 5.3 sehen, dass sich schon in der  $n$ -ten Ableitung von  $u$  bei einem Sprung in  $w$  relativ große Werte abzeichnen. Wenn nun  $u^{(n)}$  bestraft wird, statt  $u^{(n+1)}$  zu beschränken, wird der Verlauf von  $w$  merklich geglättet. Das heißt, dass durch Verwendung der



**Bild 5.4:** Bei Berücksichtigung der Aktordynamik durch eine Straffunktion ergibt sich ein wesentlich glatterer Verlauf als bei einer Nebenbedingung (Beschränkung). Dies sieht man besonders deutlich an den Zeitableitungen der Krümmung [95].

Straffunktion die Ordnung des Differenzenquotienten je nach Anforderung an  $w$  um eins reduziert werden kann, was hier für die Längsaktorik vorgenommen wird.

### 5.1.3 Berechnung diskreter Zeitableitungen höherer Ordnung

In diesem Abschnitt werden die Koeffizienten  $\alpha$ ,  $\beta$  und  $\gamma$  in (5.9) bestimmt. Doch zuerst muss (5.9) wie alle Gleichungen vom Zeitbereich in den Wegbereich transformiert werden, denn der Index  $k$  in (5.9) bezieht sich auf diskrete *Zeitpunkte*. Dafür werden die diskretisierten Zeitableitungen (beliebiger Ordnung) berechnet, indem die Wegableitung über Differenzenquotienten approximiert und in den Zeitbereich transformiert wird. Zur Beschreibung der Zeitabhängigkeit wird die Transformation (3.6) auf S. 26 für  $|\kappa d| \ll 1$  und  $\cos(\psi_t + \beta) \approx 1$ , also für den Fall, dass die Fahrlinie geringe Abweichungen zur Referenzlinie hat, approximiert, sodass sich die Anzahl der Abhängigkeiten signifikant reduziert:

$$\frac{dt}{ds} = \sigma = \frac{1 - \kappa d}{v \cos(\psi_t + \beta)} \approx \frac{1}{v} \quad (5.10)$$

Während die Approximation (5.10) für das primäre Gütemaß und die System-DGLs ungeeignet ist, weil dann relevante Abhängigkeiten wegfielen, kann sie für das sekundäre Gütemaß durchaus verwendet werden. Denn i. d. R. ist  $|\psi_t + \beta| < 15^\circ \Leftrightarrow \cos(\psi_t + \beta) > 0,966$ . Der Ausdruck  $\kappa d$  hängt von der Wahl der Referenzlinie und der Fahrbahnbreite ab und kann daher auf Rennstrecken nicht pauschal abgeschätzt werden. Auf deutschen Autobahnen liegt er unter 0,0241, wie Anhang B zeigt.

Im Folgenden wird die Variable  $\xi$  stellvertretend für eine beliebige Zustandsvariable aus  $\mathbf{x}$  oder Eingangsvariable aus  $\mathbf{u}$  verwendet, und der Vektor  $\xi$  ist demnach der Vektor aller  $N$  bzw.  $(N+1)$  diskreten Werte  $\xi_k := \xi(s_k)$ . Aus (5.10) ergibt sich also die Zeitableitung

$$\frac{d^n \xi}{dt^n} = \frac{ds^n}{dt^n} \frac{d^n \xi(s)}{ds^n} \approx v(s)^n \cdot \frac{d^n \xi(s)}{ds^n}. \quad (5.11)$$

Die Wegableitung von  $\xi$  wird durch einen Vorwärts-Differenzenquotienten approximiert, damit das Differentiationsverfahren zum verwendeten Integrationsverfahren (Explizites EULER-Verfahren) passt und keine Phasenverschiebung zwischen den einzelnen Termen auftritt. Die erste Ableitung ist somit

$$\xi'_k = \frac{\Delta \xi_k}{\Delta s_k} = \frac{\xi_{k+1} - \xi_k}{s_{k+1} - s_k}. \quad (5.12)$$

Der Differenzenquotient (5.12) kann für alle  $k = 1, \dots, N$  als Matrix-Vektor-Produkt

$$\begin{bmatrix} \xi'_1 \\ \vdots \\ \xi'_N \end{bmatrix} = \mathbf{D} \xi \quad (5.13)$$

mit einer Sparse-, also dünn besetzten, Bandmatrix  $\mathbf{D}$  angegeben werden, deren Elemente von  $\Delta s_k$  abhängen. Differenzenquotienten höherer Ordnung können analog dazu hergeleitet werden. Die Matrix  $\mathbf{D}$  wird im Folgenden mit einem Index  $n$  versehen, welcher die Ordnung angibt, z. B.  $\mathbf{D}_2$  für die 2. Ableitung nach  $s$ . Hiermit kann die Zeitableitung  $n$ -ter Ordnung über

$$\begin{bmatrix} \frac{\Delta^n \xi_1}{\Delta t_1^n} \\ \vdots \\ \frac{\Delta^n \xi_N}{\Delta t_N^n} \end{bmatrix} \approx \text{diag}(\mathbf{v})^n (\mathbf{D}_n \xi) \quad \text{mit} \quad \text{diag}(\mathbf{v})^n = \begin{bmatrix} v_1^n & & \\ & \ddots & \\ & & v_N^n \end{bmatrix} \quad (5.14)$$

ermittelt werden. Um die Formeln übersichtlicher zu gestalten, wird in Anlehnung an [65] folgende Notation für die elementweise Potenz eines Vektors  $\mathbf{v} \in \mathbb{R}^N$  eingeführt:

$$\mathbf{v}^{oi} := [v_1^i \quad \dots \quad v_N^i]^\top = \underbrace{\mathbf{v} \odot \mathbf{v} \odot \dots \odot \mathbf{v}}_{i\text{-mal}}, \quad i \in \mathbb{N}_0 \quad (5.15)$$

So lässt sich (5.14) kompakt als  $\mathbf{v}^{on} \odot (\mathbf{D}_n \xi)$  schreiben.

Die Koeffizienten des Vorwärts-Differenzenquotienten höherer Ordnung, d. h. die Einträge der Matrix  $\mathbf{D}$ , können über verschiedene Wege bestimmt werden. Der klassische Weg wendet (5.12) rekursiv an. Hier wird hingegen ein Ansatz gewählt, der  $\xi(s)$  in eine TAYLOR-Reihe entwickelt. Er geht auf einen Ansatz zur Diskretisierung von Randwertproblemen mit nicht-äquidistanten Gittern mittels zentraler Differenzenquotienten in [41, Kap. 7] zurück, welcher auf diese Arbeit angepasst wurde. Ein zentraler Differenzenquotient ist hier nämlich unabhängig vom Integrationsverfahren ungeeignet. Denn während der Optimierung besteht dann keine Möglichkeit, eine Oszillation mit der Abtastfrequenz zu verhindern. Die Begründung hierfür findet sich in Anhang C.

Die Koeffizienten des Differenzenquotienten sollen nun exemplarisch für die Ordnung  $n = 2$  hergeleitet werden, indem die Ableitung über die Ansatzfunktion

$$\frac{\Delta^2 \xi_k}{\Delta s_k^2} = \xi_k^{(2)} = \alpha_k \xi_k + \beta_k \xi_{k+1} + \gamma_k \xi_{k+2} \quad (5.16)$$

ermittelt wird, welche wir schon aus (5.9) kennen. Alle Variablen  $\xi_{k+i} = \xi(s_{k+i})$ ,  $i = 0, \dots, n$ , aus (5.16) werden jeweils um  $s = s_k$  herum in eine TAYLOR-Reihe  $n$ -ter Ordnung

$$\xi(s) = \xi(s_k) + \frac{s - s_k}{1!} \xi^{(1)}(s_k) + \frac{(s - s_k)^2}{2!} \xi^{(2)}(s_k) + \dots + \frac{(s - s_k)^n}{n!} \xi^{(n)}(s_k) + R_n(s) \quad (5.17)$$

entwickelt, sodass sich mit  $h_k := s_{k+1} - s_k$  und  $n = 2$

$$\xi_k = \xi_k + R_2(s_k) \quad (5.18a)$$

$$\xi_{k+1} = \xi_k + h_k \xi_k^{(1)} + \frac{h_k^2}{2} \xi_k^{(2)} + R_2(s_{k+1}) \quad (5.18b)$$

$$\xi_{k+2} = \xi_k + (h_k + h_{k+1}) \xi_k^{(1)} + \frac{(h_k + h_{k+1})^2}{2} \xi_k^{(2)} + R_2(s_{k+2}) \quad (5.18c)$$

ergibt. Setzt man (5.18) unter Vernachlässigung der Restglieder in (5.16) ein, erhält man:

$$\begin{aligned} \xi_k^{(2)} &= \underbrace{(\alpha_k + \beta_k + \gamma_k)}_{\stackrel{!}{=0}} \xi_k + \dots \\ &\quad \underbrace{(\beta_k h_k + \gamma_k (h_k + h_{k+1}))}_{\stackrel{!}{=0}} \xi_k^{(1)} + \dots \\ &\quad \underbrace{\frac{1}{2}(\beta_k h_k^2 + \gamma_k (h_k + h_{k+1})^2)}_{\stackrel{!}{=1}} \xi_k^{(2)} \end{aligned} \quad (5.19)$$

Um (5.19) zu erfüllen, müssen alle Summanden außer  $\xi_k^{(2)}$  verschwinden, woraus sich das LGS

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & h_k & h_k + h_{k+1} \\ 0 & \frac{1}{2}h_k^2 & \frac{1}{2}(h_k + h_{k+1})^2 \end{bmatrix} \begin{bmatrix} \alpha_k \\ \beta_k \\ \gamma_k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \Rightarrow \begin{cases} \alpha_k &= \frac{2}{h_k(h_k + h_{k+1})} \\ \beta_k &= -\frac{2}{h_k h_{k+1}} \\ \gamma_k &= \frac{2}{h_{k+1}(h_k + h_{k+1})} \end{cases} \quad (5.20)$$

ergibt. Im Gegensatz zur klassischen rekursiven Berechnung sind die Zähler der Koeffizienten in (5.20) unabhängig von  $h$  und die Terme allgemein kürzer, was eine schnellere Berechnung erlaubt. Der Unterschied zur rekursiven Berechnung ist in der Lösungstrajektorie allerdings marginal.

Zusammengefasst ergibt sich für (5.13) eine Bandmatrix, deren Diagonalen die Koeffizienten beinhalten, sodass die Anzahl ihrer Bänder eins höher ist als die Ordnung des Differenzenquotienten. Sie lautet für eine offene Strecke bzw. einen Rundkurs

$$\mathbf{D}_{2,Ab} = \begin{bmatrix} \alpha_1 & \beta_1 & \gamma_1 & & \\ & \ddots & \ddots & \ddots & \\ & & \alpha_{N-2} & \beta_{N-2} & \gamma_{N-2} \\ & & & 0 & 0 \\ & & & & 0 \end{bmatrix}, \quad \mathbf{D}_{2,Rk} = \begin{bmatrix} \alpha_1 & \beta_1 & \gamma_1 & & \\ & \ddots & \ddots & \ddots & \\ & & \alpha_{N-2} & \beta_{N-2} & \gamma_{N-2} \\ \gamma_{N-1} & & & \alpha_{N-1} & \beta_{N-1} \\ \beta_N & \gamma_N & & & \alpha_N \end{bmatrix}.$$

Bei einer offenen Strecke können die letzten Vorwärts-Differenzenquotienten nicht berechnet werden, was aber in der Praxis keine negativen Auswirkungen hat, da die letzten Punkte bei einer zyklischen Planung niemals abgefahren werden sollten. Würden sie verwendet, weil bis dahin keine neue Planung erfolgt ist, ist es für eine Fortführung der automatisierten Fahrt i. d. R. ohnehin zu spät, was mit einem Systemausfall gleichzusetzen ist. Bei einer Rundkurs-Planung ist es hingegen wichtig, dass der Nulldurchgang exakt genauso geplant wird, wie der Rest der Strecke. Dafür sei in (5.20) bei Rundkursen  $h_{N+m} := h_m$ ,  $m = 1, \dots, n-1$ , definiert.

Für die quadratische Gütefunktion gemäß (5.9) ergibt sich nun folgende Form (i. d. R. ist hierbei  $i \equiv n$ , später wird jedoch der besondere Fall  $i \neq n$  vorkommen):

$$\begin{aligned} J_{\mathbf{D}_n} &= \frac{1}{2} \sum_{k=1}^N \left( v_k^i \frac{\Delta^n \xi_k}{\Delta s_k^n} \right)^2 = \frac{1}{2} (\mathbf{v}^{\circ i} \odot (\mathbf{D}_n \boldsymbol{\xi}))^\top (\mathbf{v}^{\circ i} \odot (\mathbf{D}_n \boldsymbol{\xi})) \\ &= \frac{1}{2} \boldsymbol{\xi}^\top \mathbf{D}_n^\top \text{diag}(\mathbf{v})^{2i} \mathbf{D}_n \boldsymbol{\xi} \end{aligned} \quad (5.21)$$

Die Gleichung ähnelt äußerlich stark der bekannten *quadratischen Form*  $J = \boldsymbol{\xi}^\top \mathbf{Q} \boldsymbol{\xi}$ , jedoch ist hier  $\mathbf{Q}(\mathbf{v}) = \mathbf{D}_n^\top \text{diag}(\mathbf{v})^{2i} \mathbf{D}_n \neq \text{const}$  von einem weiteren Zustand abhängig. Ob man dabei rein formal das Quadrat vom Zustand  $\boldsymbol{\xi}$  oder – wie hier – von der Geschwindigkeit  $v^i$  jeweils mit  $\mathbf{D}_n$  multipliziert zur Matrix  $\mathbf{Q}$  zusammenfasst, ist irrelevant. Die zusätzliche Abhängigkeit hat aber zur Folge, dass die Berechnung des Gradienten und der HESSE-Matrix von  $J_{\mathbf{D}_n}$  durchaus nicht trivial ist, vor allem wenn die Matrix-Struktur beibehalten werden soll. Die Ableitungen sind:

$$\frac{\partial J_{\mathbf{D}_n}}{\partial \boldsymbol{\xi}} = \mathbf{D}_n^\top (\mathbf{v}^{\circ 2i} \odot (\mathbf{D}_n \boldsymbol{\xi})) \quad (5.22a)$$

$$\frac{\partial J_{\mathbf{D}_n}}{\partial \mathbf{v}} = i \mathbf{v}^{\circ(2i-1)} \odot (\mathbf{D}_n \boldsymbol{\xi})^{\circ 2} \quad (5.22b)$$

$$\frac{\partial^2 J_{\mathbf{D}_n}}{\partial \boldsymbol{\xi} \partial \boldsymbol{\xi}^\top} = \mathbf{D}_n^\top \text{diag}(\mathbf{v})^{2i} \mathbf{D}_n \quad (5.22c)$$

$$\frac{\partial^2 J_{\mathbf{D}_n}}{\partial \mathbf{v} \partial \mathbf{v}^\top} = i(2i-1) \text{diag}(\mathbf{v}^{\circ(2i-2)} \odot (\mathbf{D}_n \boldsymbol{\xi})^{\circ 2}) \quad (5.22d)$$

$$\frac{\partial^2 J_{\mathbf{D}_n}}{\partial \boldsymbol{\xi} \partial \mathbf{v}^\top} = 2i \mathbf{D}_n^\top \text{diag}(\mathbf{v}^{\circ(2i-1)} \odot (\mathbf{D}_n \boldsymbol{\xi})) \quad (5.22e)$$

$$\frac{\partial^2 J_{\mathbf{D}_n}}{\partial \mathbf{v} \partial \boldsymbol{\xi}^\top} = 2i \text{diag}(\mathbf{v}^{\circ(2i-1)} \odot (\mathbf{D}_n \boldsymbol{\xi})) \mathbf{D}_n \quad (5.22f)$$

In den tatsächlich verwendeten Gütemaßen entstehen meist etwas komplexere Ausdrücke, welche dann mit Hilfe dieser allgemeinen Formeln hergeleitet werden können. Die Ergebnisse aus diesem Abschnitt werden verwendet, um in Abschnitt 5.2.2 die Gütemaße konkret zu formulieren.

## 5.2 Gütemaß der Trajektorienplanung

Nachdem in den vergangenen Abschnitten die Vorbereitungen zur Formulierung des Gütemaßes getroffen wurden, wird es nun in der Form (5.1a) angegeben. Die dazugehörigen Gradienten und HESSE-Matrizen sind in Anhang D.1 aufgeführt. Die Gütefunktion besteht aus 3 Summanden:

1. Dem primären Gütemaß  $J_t$  für die Zeitoptimalität als Diskretisierung von (3.2):

$$J_t = \sum_{k=1}^N h_k \cdot \sigma_k(\mathbf{x}_k) \quad (5.23)$$

2. Zwei sekundären Gütemaßen, die in den folgenden Abschnitten beschrieben werden:

- i. Das Gütemaß  $J_{\text{GNB}}$ , das GNB in Form einer weichen Straffunktion abbildet, wie es allgemein in (4.27) auf S. 56 und Bild 4.3(a) auf S. 58 beschrieben ist.
- ii. Das Gütemaß  $J_{\text{UNB}}$ , unter welchem alle Güteterme subsumiert werden, welche die Form der Trajektorie beeinflussen. Hierbei handelt es sich im Gegensatz zu den „weichen GNB“ in (2.i) um „weiche UNB“. Dazu gehört auch die Straffunktion (5.9) für die Aktordynamik.

*Wann ist es überhaupt sinnvoll, eine NB im Gütemaß abzubilden?*

Nebenbedingungen, die strikt eingehalten werden sollen, wie z. B. die Systemgleichung oder der Kammsche Kreis, sollten nicht als Güteterm formuliert sein. Denn so eine Straffunktion muss im unzulässigen Bereich viel größere Werte als im zulässigen Bereich annehmen, damit eine Überschreitung quasi ausgeschlossen ist. Bei UNB muss sie zudem im zulässigen Bereich flach sein, um das primäre Gütemaß nicht zu beeinflussen. Weil sie dadurch zwangsläufig extreme Änderungen im Gradienten aufweist, wird die Liniensuche und somit die Konvergenz negativ beeinflusst. Eine Barriere-Funktion stellt erst recht keine gute Alternative dar: Da IPOPT von Hause aus eine Update-Strategie des Barriere-Parameters mitbringt (s. Kapitel 4), ist eine solche fixe Barriere-Funktion meist ineffizienter. Dies betrifft insbesondere Box-Restriktionen, wie den Fahrbahnrand.

Ist hingegen eine gewisse Überschreitung der Grenze zulässig, wobei sie aber möglichst gering ausfallen soll, bietet sich ein Strafterm anstelle einer NB an. Wird dieser geschickt formuliert, kann er deutliche Performance-Vorteile gegenüber einer NB bringen, da durch den Strafterm die Dimension des Optimierungsproblems nicht anwächst (vgl. hierfür das *primal-duale System* (4.38) auf S. 60, in welchem für jede NB  $c_i(\mathbf{x}_{\text{opt}})$  eine weitere Zeile entsteht).

### 5.2.1 Sekundäres Gütemaß: Straffunktion für Gleichungsnebenbedingungen

Die Komplementärbedingung für Motor- und Bremskraft nach (3.40) auf S. 39 ist eine weiche GNB. Sie wird wie alle sekundären Gütemaße mit einem Faktor  $\lambda > 0$  gegenüber dem primären

Gütemaß gewichtet:

$$J_{\text{GNB}} = \sum_{k=1}^N \Phi_{\text{MBK}}(\mathbf{u}_k) \quad \text{mit} \quad \Phi_{\text{MBK}}(\mathbf{u}_k) = \frac{\lambda_{\text{MBK}}}{2} (u_{\text{M},k} u_{\text{B},k})^2 \quad (5.24)$$

In der Regel ist  $\Phi_{\text{MBK}}$  per se sehr klein, weil es selten zeitoptimal ist, Antrieb und Bremse gleichzeitig zu verwenden. Jedoch ist die Motorkraftverteilung  $k_{\text{MV}}$  ungleich der Bremskraftverteilung  $k_{\text{BV}}$ , sodass die simultane Verwendung von Antrieb und Bremse die resultierende Längskraftverteilung beeinflusst und so einen zusätzlichen Freiheitsgrad für die Optimierung schafft. Deswegen könnte eine zeitoptimale Lösung ohne Berücksichtigung von (5.24) gerade in dynamischen Kurven diesen unerwünschten Effekt aufweisen. Allerdings stellen niedrige Werte  $\Phi_{\text{MBK}} \gtrsim 0$ , die sich durch die Numerik ergeben können, kein Problem dar, weil sie den Trajektorienverlauf nur marginal beeinflussen.

### 5.2.2 Sekundäres Gütemaß: Straffunktion für Ungleichungsnebenbedingungen

Die Sekundär-Gütemaße für UNB lassen sich in zwei Unterkategorien aufteilen: Zum einen die bereits hinlänglich beschriebene Aktordynamik (Längskraft und Lenkwinkel) und zum anderen Terme zur Beeinflussung der Trajektorien-Form (Krümmungsverlauf). Da ihr struktureller Aufbau und ihre Wirkungsweise sehr ähnlich sind, werden sie in diesem Abschnitt gemeinsam behandelt.

#### Längskraft (Ruck)

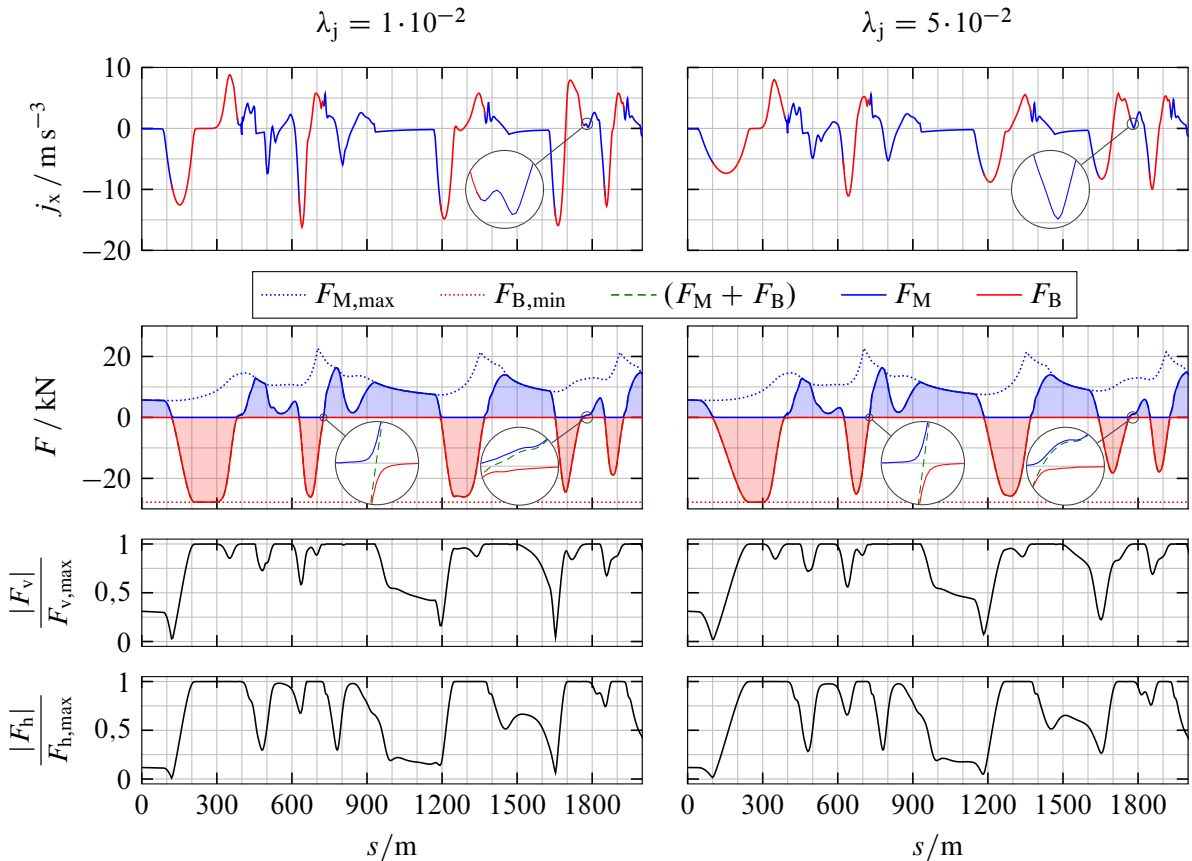
Der erste Güteterm verwendet die Modell-Eingangsgroßen  $F_{\text{M}}$  und  $F_{\text{B}}$ , wobei aufgrund der Komplementärbedingung (3.40) auf S. 39 lediglich die Summe  $F_{\text{M}} + F_{\text{B}}$  relevant ist. Für die Längsdynamik ist es bei Verwendung einer Straffunktion ausreichend, die erste Ableitung zu verwenden. Sie ist bei Geradeausfahrt proportional zum Längsruck

$$j_x \approx \frac{1}{m} \frac{d(F_{\text{M}} + F_{\text{B}})}{dt}. \quad (5.25)$$

So ergibt sich unter Verwendung von (5.21) das Gütemaß

$$J_j = \frac{\lambda_j}{2} \frac{1}{m} \left( \mathbf{v} \odot (\mathbf{D}_1(\mathbf{u}_{\text{M}} + \mathbf{u}_{\text{B}})) \right)^{\top} \left( \mathbf{v} \odot (\mathbf{D}_1(\mathbf{u}_{\text{M}} + \mathbf{u}_{\text{B}})) \right), \quad (5.26)$$

wobei der Faktor zwischen  $u_{\text{M/B}}$  und  $F_{\text{M/B}}$  in dem Gewichtungsfaktor  $\lambda_j$  integriert ist. Bild 5.5 zeigt die Auswirkung von  $J_j$  anhand einer Trajektorienplanung für das *Autódromo do Algarve*, wobei zur besseren Übersicht nur der Streckenabschnitt bis vor Kurve 9 (*Craig Jones Corner*) abgebildet ist (vgl. Bild 7.3 auf S. 134). In der Simulation in Bild 5.5 (links) wird  $\lambda_j = 1 \cdot 10^{-2}$  gewählt, rechts um den Faktor 5 größer. Oben ist der Längsruck  $j_x$ , darunter sind Motorkraft  $F_{\text{M}} \geq 0$  und Bremskraft  $F_{\text{B}} \leq 0$  (durchgezogene Linien) inklusive ihrer jeweiligen physikalischen Grenzen (gepunktete Linien) abgebildet. Zusätzlich ist die Gesamt-Reifenlängskraft  $F_{\text{M}} + F_{\text{B}}$  gestrichelt dargestellt, jedoch sieht man sie fast nur in den Ausschnittvergrößerungen. Gemäß der farblichen

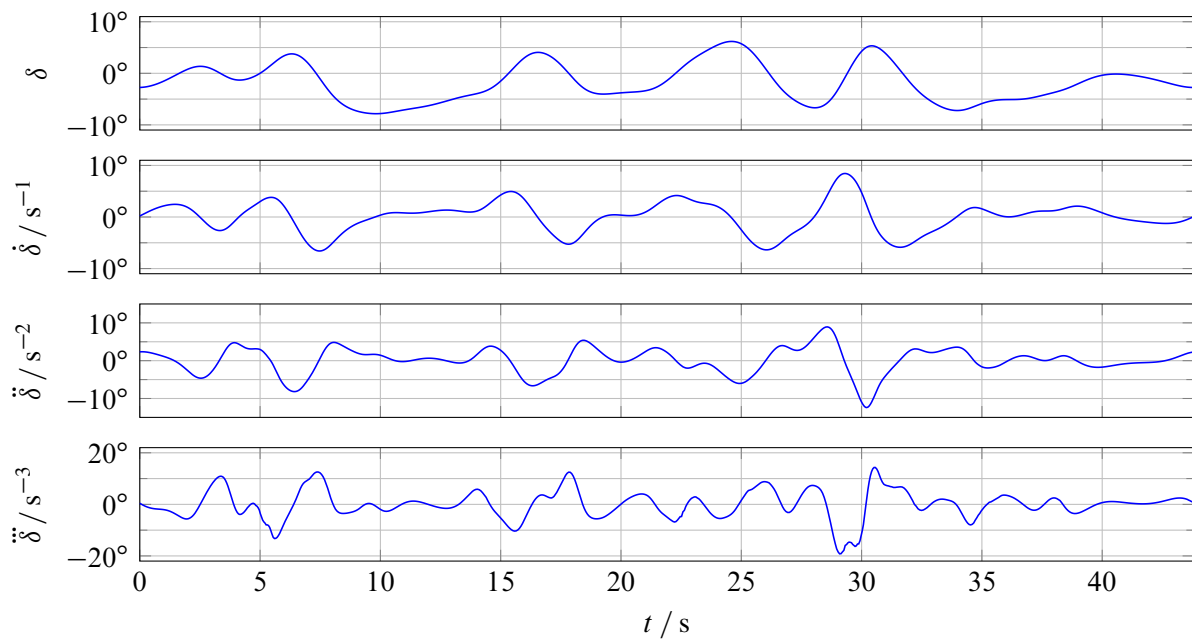


**Bild 5.5:** Über den Parameter  $\lambda_j$  wird der Längsruck (oben) beschränkt. Um den Ruck im Übergang zwischen Verzögern und Beschleunigen minimal zu halten, wird eine parallele Verwendung von  $F_M$  und  $F_B$  in geringem Maße toleriert, wie die Ausschnittvergrößerungen in der Mitte zeigen. Die beiden unteren Plots stellen die relativen Reifenhorizontalkräfte dar.

Darstellung von  $F_M$  in blau und  $F_B$  in rot ist auch  $j_x$  entsprechend eingefärbt, je nachdem ob  $(F_M + F_B)$  positiv oder negativ ist. Die Grenze  $F_{M,max}$  ergibt sich aus dem Minimum von Kraftschlussgrenze beider Reifen und dem hyperbelförmigen Lieferkennfeld (3.41).  $F_{B,min}$  wird nur durch die Kraftschlussgrenze bestimmt, da jedes straßenzugelassene Fahrzeug die Blockiergrenze erreichen kann. Die unteren Plots zeigen die relativen Reifenkräfte. Wie man an den Reifenkräften beim Beschleunigen erkennen kann, wurde hierfür ein Fzg mit Allradantrieb simuliert, dessen Motorleistung bei Geradeausfahrt in den meisten Situationen nicht ausreicht, um die Kraftschlussgrenze überschreiten zu können.

Zunächst kann man erkennen, dass der Maximal-Ruck auf der linken Seite bei  $|\hat{j}_x| \approx 15 \text{ m/s}^3$  und auf der rechten bei  $|\hat{j}_x| \approx 10 \text{ m/s}^3$  liegt, sodass eine Begrenzung des Rucks möglich ist. Bei der Analyse ist die Entstehung des Rucks interessant: Ein positiver Ruck  $j_x > 0$  bei gleichzeitiger Beschleunigung  $F_M > 0$  muss vom Antrieb aufgebracht werden, und ist somit durch die Aktorik begrenzt; ebenso ein negativer Ruck  $j_x < 0$  bei gleichzeitiger Verzögerung  $F_B < 0$  durch die Bremse. Hingegen ist ein positiver Ruck  $j_x > 0$  bei gleichzeitiger Verzögerung  $F_B < 0$  recht uninteressant, weil er beim Öffnen der Bremse entsteht, was innerhalb kürzester Zeit erfolgen kann.





**Bild 5.6:** Der Verlauf des Lenkwinkels, der sich für die Trajektorie eines Handlings-Kurses ergibt, ist so glatt, dass er mehrfach stetig differenziert werden kann.

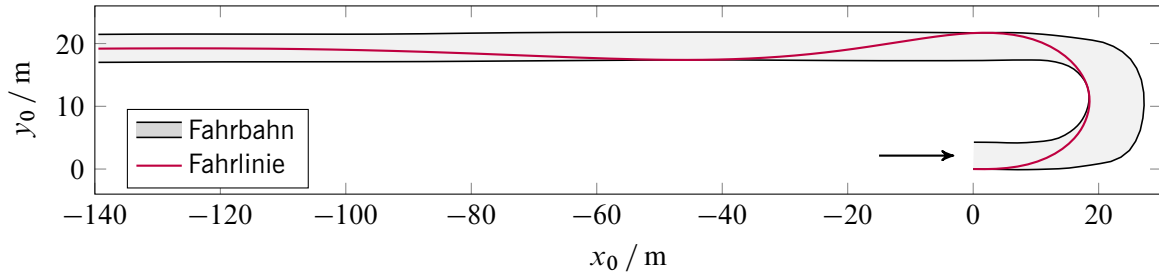
Da Motor- und Bremsruck nicht einzeln, sondern in Summe bestraft werden, entstehen in den Gesamt-Reifenkräften ( $F_M + F_B$ ), welche proportional zur SP-Beschleunigung sind, sehr glatte Übergänge zwischen Verzögern und Beschleunigen. Sie sind exemplarisch in zwei Ausschnittvergrößerungen zu sehen. Dieser Übergang wird dadurch ermöglicht, dass die Komplementärbedingung für Motor- und Bremskraft (3.39) auf S. 39 durch eine „weiche“ Straffunktion (5.24) formuliert ist und deshalb eine Verletzung der Komplementärbedingung in geringem Maß zulässt, wie man in der Vergrößerung erkennen kann. Daran sieht man, wie wichtig es ist, alle Aspekte eines Optimierungsproblems aufeinander abzustimmen und nicht nur separat zu betrachten.

Die Vergrößerung bei  $s = 1780$  m zeigt, dass die Motorkraft dort leicht stufenförmig zunimmt. Der Verlauf ist bei großem  $\lambda_j$  glatter als bei kleinem, wie man auch am Ruck (siehe Lupe) erkennen kann. Der Grund für den stufenförmigen Verlauf ist, dass die Vorderachse so weit am Limit ist, dass eine volle Beschleunigung unmöglich ist (siehe Diagramm). Je nach Gewichtungsfaktor  $\lambda_j$  wird dann ein Kompromiss zwischen niedrigem Ruck und Zeitoptimalität gewählt.

Die simulierte Rundenzeit ist durch die Ruckreduzierung kaum größer: Für den linken Plot beträgt sie für den 4,5 km langen Rundkurs  $T_{\text{lap}} = 122$  s, für den rechten ist sie 2/10 s (0,16 %) größer.

### Lenkwinkel

Der Lenkwinkel  $\delta$  soll mindestens zweimal stetig differenzierbar sein. Hierfür wird die 3. Ableitung verwendet. Zusätzlich wird auch die 2. Ableitung bestraft, da diese Kombination zu einer effektiven Unterdrückung von nieder- und höherfrequenten Schwingungen führt. Die Gütefunk-



**Bild 5.7:** Wird die Gewichtung von  $J_\delta$  zu groß gewählt, entsteht in manchen Situationen ein Überschwängen, wie es hier für eine extreme Situation dargestellt ist.

tion wird wieder in der Form von (5.21) angegeben:

$$J_\delta = \frac{\lambda_{\delta,2}}{2} (\mathbf{v}^{\circ 2} \odot (\mathbf{D}_2 \delta))^\top (\mathbf{v}^{\circ 2} \odot (\mathbf{D}_2 \delta)) + \frac{\lambda_{\delta,3}}{2} (\mathbf{v}^{\circ 3} \odot (\mathbf{D}_3 \delta))^\top (\mathbf{v}^{\circ 3} \odot (\mathbf{D}_3 \delta)) \quad (5.27)$$

In Bild 5.6 ist exemplarisch ein Lenkwinkelverlauf für einen Handlings-Kurs abgebildet. Da für die Betrachtung der Differenzierbarkeit die Zeitabhängigkeit wichtiger ist als die Ortsabhängigkeit, ist er über der Fahrzeit geplottet. Die abgebildeten Zeitableitungen wurden über Vorwärts-Differenzenquotienten berechnet, wobei für ihre Berechnung die tatsächliche Fahrzeit anstelle der Approximation (5.10) auf S. 70 verwendet wurde. Man kann sehen, dass sich ein sehr glatter, mehrfach stetig differenzierbarer Verlauf ergibt.

Die Gewichtung  $\lambda_\delta$  sollte allerdings nicht zu hoch gewählt werden, da sie *jede Änderung der Lenkrate* bestraft. In Bild 5.7 ist das Ergebnis einer bewusst übertriebenen Gewichtung dargestellt: An den Ausgang einer recht engen Kurve schließt sich eine lange Gerade an. Sie ist so lang, dass eine Verlängerung der gefahrenen Wegstrecke durch einen Seitenwechsel nur eine relativ geringe Auswirkung auf die Fahrzeit hat. Die gesamte Gerade wird ab dem Kurvenausgang, wo die Fahrlinie den rechten Rand erreicht, zum Beschleunigen verwendet. In diesem Beispiel ist die Motorleistung (147 kW) so niedrig, dass die Kraftschlussgrenze ab dem Kurvenausgang auch bei Lenkmanövern nie erreicht wird. Es ist klar, dass die Ideallinie ab dem Kurvenausgang am rechten Rand der Geraden verlaufen muss. Statt aber am Kurvenausgang den Lenkwinkel möglichst schnell in die Nullstellung zurückzuführen, damit die Linie am rechten Rand bleibt, wird der Lenkwinkel hier nur langsam reduziert, was zu dem sichtbaren Überschwängen führt. Bei einer zu hohen Gewichtung von  $J_\delta$  ist dies im Sinne des Gütemaßes optimal, denn ein schnellerer Übergang von  $\delta > 0$  zur Nullstellung  $\delta = 0$  würde zu einem Anstieg in  $|\ddot{\delta}|$  und  $|\ddot{\delta}|$  führen, was vermieden werden soll. Bei den meisten realen Streckenverläufen ist dieser Effekt nicht so deutlich sichtbar, weil entweder das primäre Gütemaß oder die Nebenbedingungen zusätzliche Lenkbewegungen verhindern.

## Krümmung

Neben den Aktorgrößen kann auch der Kurs mit sehr ähnlichen Termen beeinflusst werden, um einerseits einen glatteren und komfortableren Verlauf zu erhalten und andererseits die Ausprägung der Ideallinie dahingehend verändern zu können, dass Kurven tendenziell „runder gefahren“ werden. Der Kursverlauf und das Maß an Komfortgefühl (soweit man bei Rennlinien von Komfort

reden kann) werden maßgeblich durch den Krümmungsverlauf bestimmt. Er ist wichtiger als die Gierrate allein, da er die Information über den Kurswinkel enthält. Die Amplitude der Krümmung lässt sich schlecht als Maß heranziehen, da sie durch den Streckenverlauf bestimmt wird und bei Rennlinien ohnehin so klein wie möglich ist, siehe *krümmungsminimale Linie* in Abschnitt 2.2. Am besten geeignet sind die ersten beiden Zeitableitungen der Krümmung. Aus der Krümmung

$$\kappa_{rl} = \frac{\dot{\chi}}{v} = \frac{\dot{\beta} + \dot{\psi}}{v}, \quad \chi = \beta + \psi \quad (5.28)$$

nach (3.31) auf S. 36 ergibt sich ihre 1. Ableitung

$$\dot{\kappa}_{rl} = \frac{(\ddot{\beta} + \ddot{\psi})v - (\dot{\beta} + \dot{\psi})\dot{v}}{v^2} = \frac{\ddot{\beta} + \ddot{\psi}}{v} - \frac{\dot{\beta} + \dot{\psi}}{v} \frac{\dot{v}}{v}, \quad (5.29)$$

welche sich aus zwei Summanden zusammensetzt. Wenn man gemäß (3.3) auf S. 26  $v = ds_v/dt$  einsetzt, kann man erkennen, dass der erste Summand die Änderung der Kurswinkelrate  $\dot{\chi}$  entlang der Fahrline beschreibt, also  $d\dot{\chi}/ds_v$ . Der zweite Summand beinhaltet das Produkt aus der Änderung des Kurswinkels, also  $d\chi/ds_v$ , und der Änderung der Geschwindigkeit, also  $\dot{v}/v = dv/ds_v$ . Der erste berücksichtigt also eher die Querbewegung und der zweite eher die Längsbewegung, wobei die Krümmungsänderung gerade bei höherer Geschwindigkeit vornehmlich durch die Querbewegung beeinflusst wird. Auf Rennstrecken kann man trotz der hohen Längsbeschleunigungen von einem typischen Wert  $\frac{|\dot{v}|}{v} \ll 0,5 \text{ s}^{-1}$  ausgehen. Vernachlässigt man im Gütemaß, welches v. a. die Bahn und weniger das Geschwindigkeitsprofil glätten soll, diesen zweiten Summanden und wendet zudem die Approximation  $\dot{s} \approx v$  gemäß (5.11) auf S. 71 an, was gleichbedeutend mit  $ds_v \approx ds$  ist, ergibt sich für die 1. Ableitung der Krümmung

$$\dot{\kappa}_{rl} \approx \frac{\ddot{\beta} + \ddot{\psi}}{v} \approx v \frac{d^2\beta}{ds^2} + \frac{d\dot{\psi}}{ds}. \quad (5.30)$$

Es fällt auf, dass in (5.30) die Potenz von  $v$  um eins niedriger ist als die Ordnung der jeweiligen Weg-Ableitung. Dies liegt daran, dass die Krümmung eine geometrische (und keine fahrdynamische) Größe ist, auch wenn sie sich auf fahrdynamische Größen zurückführen lässt. Denn (5.28) lässt sich auch in der Form

$$\kappa_{rl} = \frac{d\chi}{d\mathcal{T}} \frac{d\mathcal{T}}{ds_v} = \frac{d\chi}{ds_v} \quad (5.31)$$

angeben, sodass sich die Krümmung also durch die *Wegableitung* des Kurswinkels  $\chi$  ergibt.

Die 2. Ableitung der Krümmung wird in analoger Vorgehensweise berechnet:

$$\ddot{\kappa}_{rl} = \frac{(\ddot{\beta} + \ddot{\psi})v - (\dot{\beta} + \dot{\psi})\dot{v}}{v^2} - \frac{((\ddot{\beta} + \ddot{\psi})\dot{v} + (\dot{\beta} + \dot{\psi})\ddot{v})v^2 - (\dot{\beta} + \dot{\psi})\dot{v} \cdot 2v\dot{v}}{v^4} \quad (5.32a)$$

$$= \frac{\ddot{\beta} + \ddot{\psi}}{v} - 2 \frac{(\dot{\beta} + \dot{\psi})\dot{v}}{v^2} + \underbrace{\frac{\dot{\beta} + \dot{\psi}}{v^2} \left( \frac{2\dot{v}^2}{v} - \ddot{v} \right)}_{\approx 0} \quad (5.32b)$$

$$\approx v^2 \frac{d^3\beta}{ds^3} + v \frac{d^2\dot{\psi}}{ds^2} - 2 \underbrace{\left( v \frac{d^2\beta}{ds^2} + \frac{d\dot{\psi}}{ds} \right) \frac{dv}{ds}}_{\rightarrow \text{vgl. (5.30)}} \quad (5.32c)$$

Für das Gütemaß zur Bestrafung von  $\ddot{\kappa}_{rl}$  werden nur die ersten beiden Summanden von (5.32c) verwendet, weil der Klammerausdruck bereits durch  $\dot{\kappa}_{rl}$  mit (5.30) bestraft wird. Dem liegt die vereinfachende Annahme zu Grunde, dass der Betrag einer Summe sehr klein wird, wenn der Betrag aller Summanden minimal ist. Dies ist zwar nicht exakt das Minimum der Summe, führt in dem konkreten Anwendungsfall aber zu sehr guten Ergebnissen und reduziert den Rechenaufwand immens. Zudem kann man im letzten Summanden die Multiplikation mit  $v'$  vernachlässigen, weil die Querdynamik in der Klammer mehr Minimierungspotential bietet als die Längsdynamik  $v'$ . Somit kann man  $v'$  bezüglich der Optimierung als fix ansehen und das Produkt wird dann minimal, wenn der Klammerausdruck minimal ist.

Es ergeben sich aus (5.30) und (5.32c) die beiden Gütefunktionen

$$\begin{aligned} J_{\dot{\kappa}} &= \frac{\lambda_{\dot{\kappa}}}{2} \left( \mathbf{v} \odot (\mathbf{D}_2 \boldsymbol{\beta}) + \mathbf{D}_1 \dot{\boldsymbol{\psi}} \right)^\top \left( \mathbf{v} \odot (\mathbf{D}_2 \boldsymbol{\beta}) + \mathbf{D}_1 \dot{\boldsymbol{\psi}} \right) \\ &= \frac{\lambda_{\dot{\kappa}}}{2} \left( (\mathbf{v} \odot (\mathbf{D}_2 \boldsymbol{\beta}))^\top (\mathbf{v} \odot (\mathbf{D}_2 \boldsymbol{\beta})) + (\mathbf{D}_1 \dot{\boldsymbol{\psi}})^\top (\mathbf{D}_1 \dot{\boldsymbol{\psi}}) + 2(\mathbf{v} \odot (\mathbf{D}_2 \boldsymbol{\beta}))^\top (\mathbf{D}_1 \dot{\boldsymbol{\psi}}) \right) \end{aligned} \quad (5.33)$$

und

$$\begin{aligned} J_{\ddot{\kappa}} &= \frac{\lambda_{\ddot{\kappa}}}{2} \left( \mathbf{v}^{\circ 2} \odot (\mathbf{D}_3 \boldsymbol{\beta}) + \mathbf{v} \odot (\mathbf{D}_2 \dot{\boldsymbol{\psi}}) \right)^\top \left( \mathbf{v}^{\circ 2} \odot (\mathbf{D}_3 \boldsymbol{\beta}) + \mathbf{v} \odot (\mathbf{D}_2 \dot{\boldsymbol{\psi}}) \right) \\ &= \frac{\lambda_{\ddot{\kappa}}}{2} (\mathbf{v}^{\circ 2} \odot (\mathbf{D}_3 \boldsymbol{\beta}))^\top (\mathbf{v}^{\circ 2} \odot (\mathbf{D}_3 \boldsymbol{\beta})) + \frac{\lambda_{\ddot{\kappa}}}{2} (\mathbf{v} \odot (\mathbf{D}_2 \dot{\boldsymbol{\psi}}))^\top (\mathbf{v} \odot (\mathbf{D}_2 \dot{\boldsymbol{\psi}})) + \dots \\ &\quad \lambda_{\ddot{\kappa}} (\mathbf{v}^{\circ 2} \odot (\mathbf{D}_3 \boldsymbol{\beta}))^\top (\mathbf{v} \odot (\mathbf{D}_2 \dot{\boldsymbol{\psi}})), \end{aligned} \quad (5.34)$$

worin

$$(\mathbf{v} \odot (\mathbf{D}_2 \boldsymbol{\beta}))^\top (\mathbf{D}_1 \dot{\boldsymbol{\psi}}) \equiv (\mathbf{v} \odot (\mathbf{D}_1 \dot{\boldsymbol{\psi}}))^\top (\mathbf{D}_2 \boldsymbol{\beta})$$

ineinander umgeformt werden kann. Dabei entsteht bezogen auf die allgemeine Formel (5.21) hier der Sonderfall  $i \neq n$ . Des Weiteren ist die Ableitung  $\dot{\boldsymbol{\psi}}$  bereits eine Optimierungsvariable, sodass die Ordnung des Differenzenquotienten von  $\boldsymbol{\psi}$  gegenüber  $\boldsymbol{\beta}$  um eins niedriger ist.

## 5.3 Nebenbedingungen der Trajektorienplanung

Zeitoptimale Trajektorien verlaufen zwangsläufiger Weise entlang einer physikalischen Grenze, denn wären sie nicht an einer Grenze, gäbe es ja eine Möglichkeit, die Zeit weiter zu reduzieren. Wie wir im vergangenen Abschnitt an den zusätzlichen Gütetermen gesehen haben, ist die vorliegende Trajektorienplanung zwar nicht ausschließlich zeitoptimal, dennoch sind die NB im Optimierungsproblem von großer Wichtigkeit. In Abschnitt 5.3.1 werden die Box-Restriktionen nach (5.1b) auf S. 64 behandelt, anschließend in Abschnitt 5.3.2 die nichtlinearen NB nach (5.1c). Die NB werden im Folgenden zur besseren Übersicht skalar angegeben. Sie sind aber streckenabhängig und gelten für jeden Diskretisierungspunkt. Das heißt jede skalare Variable entspricht einem Vektor mit  $N$  bzw.  $N + 1$  Elementen.

### 5.3.1 Box-Restriktionen

Die Box-Restriktionen sind:

$$\text{Motorkraft:} \quad 0 \leq F_M \leq F_{M,\max} \quad (5.35a)$$

$$\text{Bremskraft:} \quad F_{B,\min} \leq F_B \leq 0 \quad (5.35b)$$

$$\text{Lenkwinkel:} \quad -\delta_{\max} \leq \delta \leq \delta_{\max} \quad (5.35c)$$

$$\text{Geschwindigkeit:} \quad v_{\min} \leq v \leq v_{\max} \quad \text{mit } v_{\min} > 0 \quad (5.35d)$$

$$\text{Querversatz (Fahrbahnbreite):} \quad d_{\text{re}} \leq d \leq d_{\text{li}}, \quad \text{wobei i. A. } d_{\text{re}} < 0 \text{ ist} \quad (5.35e)$$

$$\text{Schwimmwinkel:} \quad -\beta_{\max} \leq \beta \leq \beta_{\max} \quad \text{mit } 0 < \beta_{\max} < \frac{\pi}{4} \quad (5.35f)$$

$$\text{Relativer Gierwinkel:} \quad -\psi_{t,\max} \leq \psi_t \leq \psi_{t,\max} \quad \text{mit } \psi_{t,\max} = \frac{\pi}{2} - \beta_{\max} \quad (5.35g)$$

Eine Begrenzung der Längskräfte  $F_M$  und  $F_B$  ist nur durch  $F_M \geq 0$  und  $F_B \leq 0$  notwendig, weil die jeweils andere Richtung durch die nichtlinearen NB für den Kammschen Kreis und das Lieferkennfeld begrenzt wird, sodass man ohne Einschränkung  $F_{M,\max} = \infty$  und  $F_{B,\min} = -\infty$  setzen kann. Allerdings lassen sich Box-Restriktionen im Optimierungsproblem viel leichter einhalten als nichtlineare NB, und die Effizienz kann steigen, wenn der Bereich einzelner Optimierungsvariablen zusätzlich eingeschränkt ist. Während sich  $F_{B,\min} = -\mu_x G$  allgemein bestimmen lässt, kann für  $F_{M,\max}$  je nach Antriebskonzept nur eine obere Schranke angegeben werden. Die Berechnung hierfür befindet sich in Anhang E.

Die Grenzen der übrigen Restriktionen ergeben sich je nach Anwendungsfall aus den technischen Daten, der Fahrbahn oder anderen Vorgaben des Nutzers. Nur (5.35f) und (5.35g) sind etwas strenger als erforderlich: Denn aus (3.6) auf S. 26 folgt lediglich die kombinierte Bedingung  $|\beta + \psi_t| < \frac{\pi}{2}$ . Doch in der Praxis stellen diese strengeren, aber einfacheren Grenzen keine relevante Einschränkung dar, solange  $\beta_{\max}$  sinnvoll gewählt wird, d. h. am besten so klein wie möglich, jedoch so groß wie nötig, damit die Optimierung nie in die Begrenzung läuft.

### 5.3.2 Nichtlineare Nebenbedingungen

Im Folgenden werden alle nichtlinearen NB  $c(\mathbf{x}_{\text{opt}})$  gemäß (5.1c) beschrieben; es sind  $5N$  Nebenbedingungen durch die Systemgleichungen,  $2N$  durch den Kammschen Kreis für Vorder- und Hinterachse,  $N$  durch die begrenzte Motorleistung und zusätzlich  $2N$  durch eine optionale Begrenzung des Schräglaufwinkels an Vorder- und Hinterachse. Die für die Optimierung wichtige JACOBI-Matrix der insgesamt  $10N$  NB ist dann

$$\nabla^T \mathbf{c} \in \mathbb{R}^{10N \times N_{\text{opt}}} \quad \text{mit} \quad N_{\text{opt}} = \dim(\mathbf{x}_{\text{opt}}) = \begin{cases} 8N, & \text{Rundkurs,} \\ 8N + 5, & \text{offene Strecke.} \end{cases} \quad (5.36)$$

Ihre Struktur, die sich im Folgenden anhand der einzelnen NB ergeben wird, ist exemplarisch für einen Rundkurs mit  $N = 5$  in Bild 5.8 abgebildet.

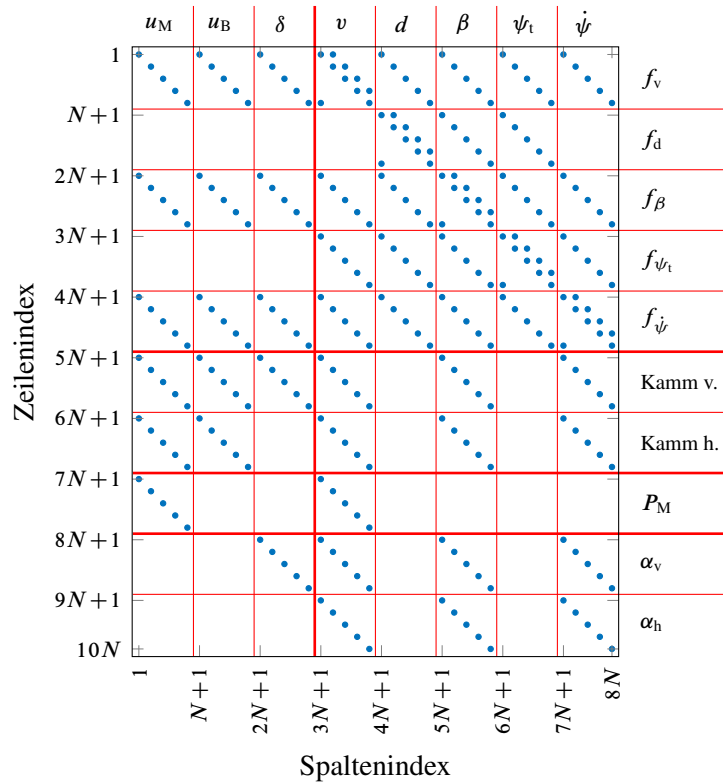


Bild 5.8: Struktur der JACOBI-Matrix der Nebenbedingungen für einen Rundkurs

### Systemgleichungen

Die Systemgleichungen (5.4) auf S. 66 führen zu den NB:

$$\mathbf{0} \leq \mathbf{x}_k - \mathbf{x}_{k+1} + \mathbf{f}_{\text{sys},k}(\mathbf{u}_k, \mathbf{x}_k) \leq \mathbf{0}, \quad k = 1, \dots, N \quad (5.37)$$

$$\text{mit } \mathbf{f}_{\text{sys},k}(\mathbf{u}_k, \mathbf{x}_k) = \begin{bmatrix} f_{v,k}(\mathbf{u}_k, \mathbf{x}_k) \\ f_{d,k}(\mathbf{u}_k, \mathbf{x}_k) \\ f_{\beta,k}(\mathbf{u}_k, \mathbf{x}_k) \\ f_{\psi_t,k}(\mathbf{u}_k, \mathbf{x}_k) \\ f_{\dot{\psi},k}(\mathbf{u}_k, \mathbf{x}_k) \end{bmatrix} = h_k \cdot \sigma_k(\mathbf{x}_k) \cdot \begin{bmatrix} \dot{v}_k(\mathbf{u}_k, \mathbf{x}_k) \\ \dot{d}_k(\mathbf{u}_k, \mathbf{x}_k) \\ \dot{\beta}_k(\mathbf{u}_k, \mathbf{x}_k) \\ \dot{\psi}_{t,k}(\mathbf{u}_k, \mathbf{x}_k) \\ \ddot{\psi}_k(\mathbf{u}_k, \mathbf{x}_k) \end{bmatrix},$$

wobei bei Rundkursen  $\mathbf{x}_{N+1} \equiv \mathbf{x}_1$  dieselbe Optimierungsvariable ist. Die NB werden nach Zuständen sortiert zu einem Vektor  $\mathbf{c}_{\text{sys}} \in \mathbb{R}^{5N}$  zusammengefasst. Aufgrund dieser Strukturierung ist die JACOBI-Matrix eine Blockmatrix, deren Blöcke aus  $(N \times N)$ - bzw.  $(N \times (N+1))$ -Untermatrizen bestehen. Die Differenz der Zustände  $(\mathbf{x}_k - \mathbf{x}_{k+1})$  an zwei aufeinanderfolgenden Punkten werde in  $\mathbf{c}_{\text{sys}}$  durch einen Summanden  $\mathbf{c}_{\text{sys},\Delta x} \in \mathbb{R}^{5N}$  beschrieben, dessen JACOBI-Matrix

$$\nabla^T \mathbf{c}_{\text{sys},\Delta x} = \begin{bmatrix} \mathbf{0} \dots \mathbf{0} & \mathbf{M} & & & \mathbf{0} \\ \mathbf{0} \dots \mathbf{0} & & \mathbf{M} & & \\ \mathbf{0} \dots \mathbf{0} & & & \mathbf{M} & \\ \mathbf{0} \dots \mathbf{0} & & & & \mathbf{M} \\ \mathbf{0} \dots \mathbf{0} & \mathbf{0} & & & \mathbf{M} \end{bmatrix} \quad (5.38)$$

$\underbrace{\hspace{10em}}_{5N \times 3N} \quad \underbrace{\hspace{10em}}_{5N \times 5N \text{ bzw. } 5N \times (5N+5)}$

$$\text{mit } \mathbf{M}_{\text{Rundkurs}} = \underbrace{\begin{bmatrix} 1 & -1 & & & \\ & \ddots & \ddots & & \\ & & 1 & -1 & \\ -1 & & & & 1 \end{bmatrix}}_{N \times N} \quad \text{bzw.} \quad \mathbf{M}_{\text{offen}} = \underbrace{\begin{bmatrix} 1 & -1 & & & \\ & \ddots & \ddots & & \\ & & 1 & -1 & \end{bmatrix}}_{N \times (N+1)}$$

ist. Nur in diesem Teil der JACOBI-Matrix kommen Untermatrizen mit Nebendiagonalen vor. Alle übrigen Terme führen nur zu Hauptdiagonalen in den Untermatrizen (vgl. Bild 5.8). Durch das verwendete explizite EULER-Verfahren sind die Nebendiagonalelemente von  $\mathbf{M}$  konstant, sodass in der HESSE-Matrix keine Nebendiagonale an dieser Stelle vorkommt. Dieser Vorteil gegenüber einem impliziten Verfahren ist der Hauptgrund für den Einsatz des expliziten Verfahrens.

Die noch fehlende Ableitung von  $\mathbf{f}_{\text{sys}}$  ist dann (in skalarer statt vektorieller Notation)

$$\nabla^T \mathbf{f}_{\text{sys}} = \nabla^T \begin{bmatrix} f_v \\ f_d \\ f_\beta \\ f_{\psi_t} \\ f_{\dot{\psi}} \end{bmatrix} = \begin{bmatrix} h \sigma \nabla^T \dot{v} + h \dot{v} \nabla^T \sigma \\ h \sigma \nabla^T \dot{d} + h \dot{d} \nabla^T \sigma \\ h \sigma \nabla^T \dot{\beta} + h \dot{\beta} \nabla^T \sigma \\ h \nabla^T (\sigma \dot{\psi} - \kappa) \\ h \sigma \nabla^T \ddot{\psi} + h \ddot{\psi} \nabla^T \sigma \end{bmatrix}. \quad (5.39)$$

Darin berechnet sich  $\nabla \sigma$  gemäß (D.3) in Anhang D.1 und die Ableitungen der restlichen Systemgleichungen stehen in Anhang D.2.1. Durch die Transformation vom Zeit- in den Wegbereich, also der Multiplikation der zeitlichen DGLs mit  $\sigma$ , entstehen zusätzliche Abhängigkeiten der Dynamik-Gleichungen  $f_v$ ,  $f_\beta$  und  $f_{\dot{\psi}}$  von den kinematischen Größen  $d$ ,  $\psi_t$  und  $v$ , wie man auch in Bild 5.8 sehen kann. Das zeigt den zusätzlichen mathematischen Aufwand, der in Abschnitt 2.3 als Nachteil der Transformation angeführt ist.

Zur Berechnung der HESSE-Matrizen aller NB können Terme der ersten Ableitungen wiederverwendet werden, was nicht nur den Implementierungsaufwand, sondern auch die Rechenzeit reduziert. Zur Berechnung der vektoriellen Ketten- und Produktregel können folgende allgemeingültige Formeln angewendet werden:

Für drei Funktionen  $f, g, h : \mathbb{R}^N \rightarrow \mathbb{R}$  gilt

$$\nabla^2 f := \nabla^T (\nabla f), \quad (5.40a)$$

$$\nabla^T (f \nabla g) = \nabla g \cdot \nabla^T f + f \nabla^2 g, \quad (5.40b)$$

$$\nabla^2 (fg) = f \nabla^2 g + \underbrace{\nabla f \cdot \nabla^T g + \nabla g \cdot \nabla^T f}_{= \nabla f \cdot \nabla^T g + (\nabla f \cdot \nabla^T g)^T \rightarrow \text{Symmetrie}} + g \nabla^2 f \quad (5.40c)$$

sowie der Spezialfall von (5.40b)

$$\nabla^T \left( \frac{f}{h} \nabla g \right) = \nabla g \cdot \nabla^T \left( \frac{f}{h} \right) + \frac{f}{h} \nabla^2 g = \frac{1}{h} \nabla g \cdot \nabla^T f + \frac{f}{h} \left( -\frac{1}{h} \nabla g \cdot \nabla^T h + \nabla^2 g \right). \quad (5.40d)$$

Dabei ist zu beachten, dass natürlich  $\nabla f \cdot \nabla^\top f \neq \nabla^2 f$  ist. Nun ergeben sich die äußeren Ableitungen der Systemgleichungen, in welchen sich die Abhängigkeiten von  $\sigma$  fortsetzen:

$$\nabla^2 f_v = h \left( \sigma \nabla^2 \dot{v} + \nabla \sigma \nabla^\top \dot{v} + \nabla \dot{v} \nabla^\top \sigma + \dot{v} \nabla^2 \sigma \right) \quad (5.41a)$$

$$\nabla^2 f_d = h \nabla^2 \left( (1 - \kappa d) \tan(\psi_t + \beta) \right) \quad (5.41b)$$

$$\nabla^2 f_\beta = h \left( \sigma \nabla^2 \dot{\beta} + \nabla \sigma \nabla^\top \dot{\beta} + \nabla \dot{\beta} \nabla^\top \sigma + \dot{\beta} \nabla^2 \sigma \right) \quad (5.41c)$$

$$\nabla^2 f_{\psi_t} = h \left( \sigma \nabla^2 \dot{\psi}^0 + \nabla \sigma \nabla^\top \dot{\psi} + \nabla \dot{\psi} \nabla^\top \sigma + \dot{\psi} \nabla^2 \sigma \right) \quad (5.41d)$$

$$\nabla^2 f_{\dot{\psi}} = h \left( \sigma \nabla^2 \ddot{\psi} + \nabla \sigma \nabla^\top \ddot{\psi} + \nabla \ddot{\psi} \nabla^\top \sigma + \ddot{\psi} \nabla^2 \sigma \right) \quad (5.41e)$$

Die inneren Ableitungen befinden sich im Anhang: Die Ableitungen von  $\sigma$  sind in Anhang D.1 angegeben, die der Systemgleichungen in Anhang D.2.1 und Anhang D.3.1.

### Kammscher Kreis

Der Kammsche Kreis nach (2.3) auf S. 15 führt mit der Reifenkennlinie aus (3.23) auf S. 32 respektive (3.19) jeweils für die Vorder- und Hinterachse zu der NB

$$F_{\text{norm}}^2 = \left( \frac{F_{x_r}}{F_{\text{max}, x_r}} \right)^2 + F_{y_r, \text{rel}}^2 \leq 1 \quad \text{mit} \quad F_{y_r, \text{rel}} = \sin \left( C \arctan(B\alpha) \right). \quad (5.42)$$

Darin ist  $F_{x_r}$  die Reifenlängskraft (3.57) auf S. 43 und  $\alpha$  der Schräglaufwinkel (3.36) auf S. 37. Die maximale Reifenkraft  $F_{\text{max}, x_r}$  ergibt sich aus der Achslastverteilung (3.51) und hängt von den Längskräften ab, wenn wie hier  $h_{\text{sp}} > 0$  ist. Die ersten und zweiten Ableitungen von (5.42) befinden sich in Anhang D.2.2 bzw. Anhang D.3.2.

### Maximale Längskraft durch die Motorleistung

Bei Elektrofahrzeugen ist aufgrund des nahezu konstanten Drehmoments über einen weiten Geschwindigkeitsbereich die Box-Restriktion (5.35a) zur Begrenzung der Längskraft maßgebend. Bei Fahrzeugen mit Verbrennungsmotor ist hingegen die verfügbare Leistung nahezu konstant, sodass das maximal abrufbare Moment, also auch die maximal verfügbare Längskraft, geschwindigkeitsabhängig ist. Ausgehend von einem idealen Lieferkennfeld [15, Kap. 3.2.1] ergibt sich gemäß (3.41) auf S. 39 die nichtlineare NB (Ableitungen in Anhang D.2.3 bzw. Anhang D.3.3)

$$F_M v \leq P_M. \quad (5.43)$$

Wenn der Kammsche Kreis nicht wie hier die Reifenkräfte an beiden Achsen einzeln begrenzt, sondern wie in [14, 50] die Gesamtbeschleunigung im SP, kann (5.43) in den Kammschen Kreis integriert werden, indem er geschwindigkeitsabhängig an der Oberseite abgeflacht wird, d. h. die positive Längsbeschleunigung begrenzt wird. Die Geschwindigkeitsabhängigkeit wird in [14, 50] durch die Annahme, dass der Arbeitspunkt während einer Optimierung konstant ist, aufgehoben.



Diese Annahme soll hier nicht getroffen werden. Unabhängig davon lassen sich die beiden UNB hier aber auch deshalb nicht zusammenfassen, weil der Einfluss der Motorkraft auf die relative Reifenkraft an den einzelnen Achsen wegen der dynamischen Achslastverteilung nicht im Voraus bekannt ist, wie in Anhang E gezeigt wird.

### Schräglaufwinkel

Eine Begrenzung des Schräglaufwinkels  $\alpha$  ist nicht notwendig, da sich durch die Reifenkennlinie, die in (5.42) eingeht, per se eine Begrenzung

$$-\alpha_{\max} \leq \alpha \leq \alpha_{\max} \quad (5.44)$$

ergibt, sodass  $\alpha$  stets betragsmäßig kleiner ist als der Wert  $\alpha_{\max} = s_{\text{krit}}$  nach (3.24) auf S. 32, bei dem die Reifenkennlinie ihr Maximum hat. Jedoch hat (5.42) viele numerisch komplexe Abhängigkeiten, während (5.44) nur wenige Abhängigkeiten aufweist, wie man an seinem Gradienten (D.15c) bzw. (D.15d) in Anhang D.2.4 sieht. Daher kann diese zusätzliche NB (5.44) gerade für Trajektorien, bei denen die UNB (5.42) häufig aktiv ist, die Konvergenz des Optimierungsproblems verbessern, auch wenn sich dadurch seine Dimension erhöht.

Jedoch muss bei einer mehr oder weniger redundanten NB besonderes Augenmerk darauf gerichtet werden, ob sie linear unabhängig zu allen anderen NB ist. Die Überprüfung auf lineare Unabhängigkeit ist generell wichtig, um die notwendige *linear independence constraint qualification* (LICQ), die in Abschnitt 4.3.1 behandelt wurde, zu erfüllen. Sie für alle  $\mathbf{x}_{\text{opt}} \in \mathbb{R}^{N_{\text{opt}}}$  zu erfüllen, ist meist unmöglich. Daher gibt es im Solver entsprechende Strategien, auf die in Abschnitt 4.4.1 eingegangen wurde. Jedoch sollte sich die LICQ durch die Struktur generell erfüllen lassen. An Bild 5.8 kann man sehen, dass alle Zeilen der JACOBI-Matrix linear unabhängig sind, also die LICQ erfüllt ist, wenn es keine besonders ungünstigen Nullspalten in der Matrix gibt.

## 5.4 Implementierung

Im vorherigen Abschnitt wurde das gesamte Problem der Trajektorienplanung auf mathematischer Ebene so definiert und formuliert, dass es mit dem gewählten Algorithmus IPOPT gelöst werden kann. Im nächsten Schritt muss es für die Zielhardware implementiert werden. Bereits die anfängliche Problemformulierung und Modellierung wurde von dem Ziel einer kurzen Rechenzeit beeinflusst; bei der Implementierung ist dieser Aspekt ebenso wichtig. Deswegen wird zum einen auf den Einsatz von Tools zur automatischen Code-Generierung verzichtet und zum anderen besonders geeignete Parallelisierungsmethoden verwendet. Im Folgenden wird beides erläutert.

Viele Tools, die eine leichte und schnelle Implementierung von Optimierungsproblemen ermöglichen (z. B. CasADi [3] oder YALMIP [51]), bieten neben einem Interface, das die mathematische Formulierung in einer abstrakteren, oft noch kontinuierlichen, also noch nicht transkribierten Form erlaubt, auch die Option, die Ableitungen automatisch zu berechnen. Denn wie in Abschnitt 4.2.2

erklärt ist, ist neben dem Gradienten der LAGRANGE-Funktion auch die HESSE-Matrix notwendig, um die Suchrichtung mittels NEWTON-Verfahren zu bestimmen. Da gerade die symbolische Berechnung der HESSE-Matrix – wie NOCEDAL UND WRIGHT treffend beschreiben – ein „cumbersome, error-prone, and expensive process“ [58, S. 23] ist, liegt es nahe, die Ableitungen nicht analytisch, sondern entweder *numerisch* oder *automatisch* zu berechnen. In unserem Projekt hat sich jedoch gezeigt, dass bei einem derart komplexen Optimierungsproblem sowohl die Konvergenz als auch die Qualität der Lösung massiv leiden, wenn entweder die Ableitungen numerisch berechnet werden oder ein Quasi-NEWTON-Verfahren zum Einsatz kommt. Beim automatischen Differenzieren (AD) werden die Ableitungen exakt berechnet, indem die klassischen Ableitungsregeln auf jede elementare arithmetische Operation im Quellcode angewendet werden [58, Kap. 8]. Jede Variable im Code enthält eine duale Variable, die ihren Gradienten angibt, und bei jeder Rechenoperation der primären Variable mit aktualisiert wird. AD hat allerdings den Nachteil, dass Termvereinfachungen und das Speichern bereits ausgewerteter Teilterme oft auf mikroskopischer Ebene erfolgen. Eine höhere Effizienz wird erreicht, wenn größere Blöcke wiederverwendet werden können. Außerdem gehen beim AD die physikalische Bedeutung und Struktur der Gleichungen zum Teil verloren, was eine Analyse erschwert. Deswegen wird in dieser Arbeit auf den Einsatz eines AD-Verfahrens verzichtet und alle Ableitungen werden analytisch berechnet, sorgfältig vereinfacht und manuell implementiert, um die größtmögliche Effizienz zu erzielen. Eine Verifikation der ermittelten Ableitungen erfolgt mit WOLFRAM MATHEMATICA.

Da mit einer exakten HESSE-Matrix die Suchrichtung und die Liniensuche präziser bestimmt werden können als bei einem Quasi-NEWTON-Verfahren, nimmt die Konvergenz so stark zu, dass sich aufgrund der geringeren Anzahl an Iterationen ein deutlicher Rechenzeitgewinn einstellt. Allerdings ist es darüber hinaus wünschenswert, dass die exakte Berechnung innerhalb einer einzelnen Iteration auch noch deutlich schneller abläuft als die Quasi-NEWTON-Approximation. Doch das ist keine triviale Aufgabe, denn Quasi-NEWTON-Verfahren sind inzwischen sehr ausgereift. Für hohe Effizienz sind zwei Aspekte wichtig: Erstens die schon angesprochene Verwendung von Zwischenergebnissen, die sowohl auf mikroskopischer Ebene erfolgt (z. B. die Vorabberechnung von  $\sin(\delta_k) \forall k$ ) als auch auf makroskopischer Ebene (z. B. die Vorabberechnung aller Reifenkräfte). Zweitens müssen die *evaluation functions*, also die Funktionen im Quellcode, welche die Terme für Gütemaß, Gradient, NB, JACOBI- und HESSE-Matrix auswerten, so programmiert sein, dass das Potential der Ziel-Hardware-Architektur bestmöglich ausgenutzt wird.

Um die Ausführung von Funktionen zu beschleunigen, ist Parallelisierung ein naheliegender Ansatz. Doch sollte Parallelisierung nicht als Mittel eingesetzt werden, um ineffizienten Quellcode durch massiven Ressourcenverbrauch noch hinreichend schnell ausführen zu können, sondern um die Rechenlast von bestmöglich optimiertem Quellcode auf die vorhandenen Ressourcen zu verteilen. Bei paralleler Ausführung von Rechenoperationen kommen zwei Architekturen besonders häufig vor: Bei einer vollständig parallelen Ausführung auf mehreren Prozessoren spricht man von „*Multiple Instruction, Multiple Data*“ (MIMD). Hingegen führt ein datenparalleler Rechner dieselbe Operation auf einer großen Datenmenge aus, was als „*Single Instruction, Multiple Data*“ (SIMD) bezeichnet wird. Ein Beispiel ist die paarweise Addition zweier Arrays (auch „Vektoren“ genannt). Bezüglich der Prozessorarchitektur unterscheidet man klassische *SIMD-Prozes-*

soren wie GPUs, bei denen die datenparallele Ausführung auf einer sehr großen Anzahl arithmetisch-logischer Einheiten (ALU) erfolgt, von *Vektor-Prozessoren* wie den meisten CPUs, bei denen alle Operationen auf einer ALU (oder sehr wenigen ALUs) mit massiver Fließbandverarbeitung („Pipelining“) ausgeführt werden [80, S. 89 ff.]. Viele heutige CPUs, wie der hier verwendete INTEL Core-Prozessor, bieten zur Verarbeitung von SIMD-Befehlen entsprechende Befehlssatzerweiterungen zu dem x86-Befehlssatz: Die von INTEL entwickelten „Streaming SIMD Extensions“ (SSE), die sich gegen das konkurrierende Konzept „3DNow“ von AMD durchgesetzt haben, enthalten hauptsächlich Instruktionen für Gleitkommaoperationen und ermöglichen die Verwendung spezieller Vektor-Register mit kurzer Zugriffszeit in der Mikroarchitektur. Inzwischen ist die Erweiterung SSE2 für Gleitkommazahlen mit doppelter Genauigkeit (64 Bit) Standard im Intel 64- und AMD64-Befehlssatz, und auch SSE3, SSE4 und SSE4.2 sind sehr weit verbreitet. Während SSE die Zwei-Operanden-Form  $y \leftarrow x + y$  nutzen, bei welcher eines der Quellregister zwangsläufig zerstört wird, bieten die „Advanced Vector Extensions“ (AVX) eine Drei-Operanden-Form  $y \leftarrow a + b$ , wodurch Kopieroperationen eingespart werden. Spätere Erweiterungen, wie AVX2 und AVX512, erweitern den Befehlssatz bzw. die Länge der Vektorregister von 256 auf 512 Bit. Durch das Pipelining ist es also möglich,  $n$  SIMD-Instruktionen in weniger als  $n$  Taktzyklen auszuführen. Dabei ist natürlich zu berücksichtigen, dass jede Parallelisierung oder Vektorisierung Overhead verursacht, sodass sich der Zeitgewinn erst ab einer gewissen Anzahl einstellt.

Für einen Programmierer ist ein SIMD-Prozessor einem Vektor-Prozessor sehr ähnlich [80, S. 90]. Jedoch ist der Hardware-Aufwand bei einem SIMD-Prozessor aufgrund der redundanten ALUs ungleich höher, weshalb GPUs i. A. teurer sind als vergleichbare CPUs. In diesem Projekt wird eine CPU verwendet, und zwar ein INTEL Core i-Prozessor.

Bei dem hier gewählten Optimierungsansatz hat die Trajektorie von Beginn an ihre volle Länge und in jeder Iteration werden alle Punkte der Trajektorie simultan optimiert (vgl. Abschnitt 4.1). Sie wird nicht wie bei einer Dynamischen Programmierung sukzessive aufgebaut und verlängert. Während sich die Berechnung der Trajektorienschar bei Dynamischer Programmierung sehr gut parallelisieren lässt, weil viele voneinander unabhängige Daten vorliegen, liegt hier also schon eine Art Parallelisierung auf algorithmischer Ebene vor, wodurch sich im Quellcode viele Bereiche ergeben, welche aufgrund von Datenabhängigkeiten sequentiell ausgeführt werden müssen, sodass der Zeitgewinn einer Hardware-Parallelisierung geringer ausfällt. Doch gerade dieser Optimierungsansatz führt andererseits dazu, dass alle Rechenoperationen auf alle Trajektorienpunkte angewendet werden, sodass jede Operation auf  $N$  Variablen(-paare) angewendet wird. Deswegen kann in dieser Arbeit ein Großteil der Berechnungen vektorisiert werden, wofür eine CPU sehr gut geeignet ist. Voraussetzung hierfür ist, dass die Daten im Speicher zusammenhängend abgelegt sind, was hier bspw. auch durch die spezielle Sortierung der Variablen im Optimierungsvektor  $\mathbf{x}_{\text{opt}}$  gemäß Abschnitt 5.1 auf S. 65 gegeben ist. Um die SSE- und AVX-Instruktionen in C++ nutzen zu können, können viele Optimierungs-Compiler Befehlssequenzen zu einem entsprechenden Vektor-Befehl zusammenfassen. Noch effizienter ist allerdings die Verwendung entsprechender Programmbibliotheken, wie z. B. „Basic Linear Algebra Subprograms“ (BLAS). Die Signaturen der in BLAS definierten Funktionen stellen einen De-facto-Standard dar und werden von vielen Prozessorherstellern so implementiert, dass sie auf ihren Prozessoren hocheffizient ablaufen. So

bietet INTEL die „Math Kernel Library“ (MKL) an [44], die außer BLAS auch weitere Bibliotheken, wie die „Vector Math Library“ (VML) beinhaltet, und neben dem ursprünglichen FORTRAN-Interface auch ein für C++ angepasstes Interface enthält. Da IPOPT ohnehin diese Bibliotheken benötigt, bietet es sich an, sie auch für die *evaluation functions* zu nutzen.

In dieser Arbeit fallen allein in den *evaluation functions* bei jeder Iteration über  $1300 \cdot N$  Rechenoperationen an, wobei  $N$  die Anzahl der Diskretisierungspunkte ist. Bei der Online-Planung müssen somit rund 125 000 Operationen pro Iteration für die *evaluation functions* durchgeführt werden. Unabhängig von der Vektorisierung kann der Code weiter parallelisiert werden, indem mehrere Kerne der CPU genutzt werden. Eine Möglichkeit hierfür ist die „threaded“ Variante der MKL, welche die Berechnung des Vektors auf mehrere Kerne verteilt. Auch wenn der Overhead dabei recht gering ist, stellt sich allerdings erst ab einer Vektorlänge von ca. 800 Elementen ein Zeitgewinn ein [19, S. 305]. Da zumindest die Online-Planung weniger als 800 DP verwendet, ergibt sich kein Vorteil, was sich auch bei Tests in diesem Projekt gezeigt hat. Deswegen wurde der Code auf makroskopischer Ebene parallelisiert, indem möglichst viele Funktionen in wenige parallele Threads aufgeteilt wurden, die unabhängig voneinander ablaufen können. Damit hierbei keine kritischen Datenabhängigkeiten auftreten, muss das Programm sauber strukturiert sein. Es ist beinahe unmöglich, eine so effiziente Struktur mit einer Auto-Codegenerierung oder einem AD-Tool zu erzielen. Durch die Kombination aus Vektorisierung und Parallelisierung entfallen nur ca. 5 % der gesamten Rechenzeit der Optimierung auf die Berechnung aller *evaluation functions*.

## 5.5 Ergebnisse der Rundkurs-Planung

Bis hierhin wurden die einzelnen Aspekte der Trajektorienplanung anhand von Streckenabschnitten beleuchtet, wie z. B. der Einfluss gewisser Parameter. Im Folgenden soll nach einer Zusammenfassung des aktuellen Kapitels ein Gesamtbild der Trajektorienplanung vermittelt werden, indem Ideallinien für einen vollständigen Rundkurs präsentiert werden. Zu einer detaillierten Analyse der Trajektorien werden relevante physikalische Größen auf gewissen Streckenabschnitten besprochen; eine Analyse des gesamten Rundkurses würde den Rahmen der Arbeit sprengen.

### 5.5.1 Zusammenfassung

Die Formulierung des Optimierungsproblems für IPOPT baut auf den Ergebnissen der vorherigen Kapitel auf und beginnt mit seiner Transkription, also der Diskretisierung. Hierfür wird ein Ansatz gewählt, der die Vorteile von Teil- und Volldiskretisierung bestmöglich vereint: Der Zustandsverlauf wird diskretisiert, damit die Zustände als Optimierungsvariablen zur Berechnung der Gütefunktion und NB verwendet werden können. Die Integration der DGLs wird durch entsprechende GNB berücksichtigt. Dabei wird für die numerische Integration das explizite EULER-Verfahren verwendet. Die Stellgrößen des realen Systems sind jedoch nicht die Eingangsgrößen des Modells, weil dazwischen Aktoren mit unterlagerten Regelkreisen vorhanden sind. Aus Effizienzgründen werden diese Stellgrößen nicht durch weitere Optimierungsvariablen abgebildet, sondern

als Differenzenquotient der Modell-Eingangsgrößen berechnet. Daraus ergeben sich UNB, die im Optimierungsproblem als Straffunktionen formuliert werden, was nicht nur zu kürzeren Rechenzeiten führt, sondern auch zu komfortableren Trajektorien, weil schnelle Dynamiken, die nur eine geringe Zeitreduzierung erbringen, geglättet werden. Für die Berechnung des Differenzenquotienten wird anstelle eines rekursiven Verfahrens ein Ansatz mittels TAYLOR-Reihenentwicklung verwendet, sodass sich die entsprechenden Terme schneller berechnen lassen. Durch die Transformation vom Zeit- in den Wegbereich entsteht eine zusätzliche Abhängigkeit in der quadratischen Form, wodurch die Ableitungen komplizierter werden. Dennoch können sie in Form von schnell ausführbaren Matrix-Matrix- und Matrix-Vektor-Multiplikationen berechnet werden. Allgemein wird die Integration, die sich nicht analytisch berechnen lässt, numerisch ausgeführt, während die Differentiation analytisch berechnet wird, um den numerischen Aufwand gering zu halten.

Anschließend wird das vollständige Gütemaß der Trajektorienplanung dargelegt: Es besteht aus dem zeitoptimalen primären Gütemaß und weiteren sekundären Gütemaßen, die GNB und UNB durch Straffunktionen abbilden. Sie ermöglichen auch, Einfluss auf die Form der Rennlinie zu nehmen. Die NB dominieren die Lösung des Optimierungsproblems, denn wegen der Zeitoptimalität ist fast überall mindestens eine NB aktiv. Die wichtigste Rolle spielen die nichtlinearen NB für die Kraftschlussgrenze und die Motorleistung.

Nach der mathematischen Beschreibung werden einzelne Aspekte der Implementierung beleuchtet. Um auch hierbei die Rechenzeit zu minimieren, werden alle Terme analytisch berechnet, bei denen es möglich ist. Des Weiteren werden verschiedene Möglichkeiten der Parallelisierung mit GPUs und CPUs angesprochen und die hier verwendete Vektorisierung kurz beschrieben.

### 5.5.2 Simulationsergebnisse für das Autodrom Most

Die Trajektorien sind für das *Autodrom Most* berechnet, wo sie auch schon in Fahrversuchen erfolgreich getestet wurden. Die Referenzlinie, deren Berechnung Abschnitt 6.1 beschreibt, verläuft näherungsweise mittig, ist 4,2 km lang und wird an  $N = 1\,373$  DP je nach Krümmung im Abstand von  $h = 1 \dots 5$  m abgetastet. Bild 5.9 zeigt zwei Trajektorien für unterschiedliche Fahrzeuge: Fzg 1 ist aus der Kompaktklasse, hat eine Motorleistung von  $P_{\max} = 147 \text{ kW} \approx 200 \text{ PS}$  und Frontantrieb. Fzg 2 ist eine Sportlimousine mit Allradantrieb und  $P_{\max} = 411 \text{ kW} \approx 560 \text{ PS}$  Leistung. Die Rechenzeit für die Optimierung beträgt auf einer INTEL Core i7 Mobile-CPU der 4. Generation (2014) in C++ knapp 1,7 s und in MATLAB weniger als 3 s.

Die Bilder 5.10 und 5.11 zeigen jeweils beide Trajektorien, wobei die Strecke zur besseren Übersicht auf zwei Bilder aufgeteilt wurde. Die im Text besprochenen Stellen sind gelb markiert. Folgende Größen sind über der Bogenlänge der Referenzlinie dargestellt:

- In den beiden oberen Diagrammen ist die Krümmung der Ideallinien und die zeitliche Ableitung der Krümmung mittels Differenzenquotient abgebildet. Man kann erkennen, dass selbst die Ableitung der Krümmung einen sehr glatten Verlauf aufweist, ohne dass zu starkes Überschwingen auftritt, wie es in Bild 5.7 auf S. 78 gezeigt wurde.



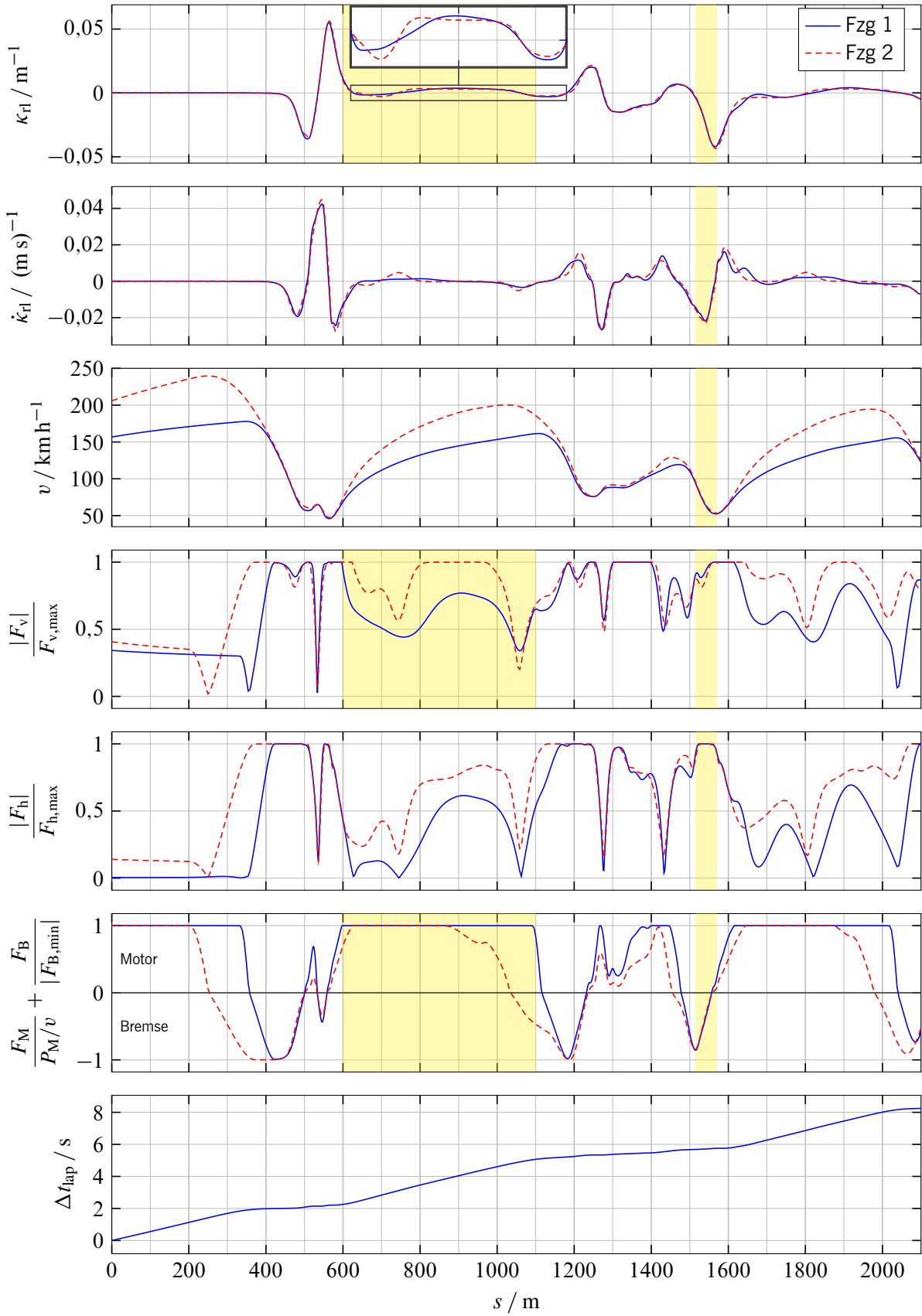
Bild 5.9: Streckenverlauf des Autodroms Most mit den Ideallinien für zwei Fahrzeuge

- Im dritten Diagramm ist die Geschwindigkeit zu sehen, an der man den Leistungsunterschied sehr deutlich erkennen kann. Fzg 1 verliert die meiste Zeit auf den geraden Stücken, während die Geschwindigkeit in den Apizes fast identisch ist. Der Leistungsunterschied ist hier nicht groß genug, um Effekte wie in Bild 2.3 auf S. 16 zu sehen, wo das hoch motorisierte Fzg engere Radien mit niedrigerer Geschwindigkeit fährt als das schwach motorisierte.
- Darunter ist der Betrag der Reifenkräfte an beiden Achsen abgebildet, welcher jeweils auf das fahrzeug- und streckenabhängige Maximum der jeweiligen Reifenkraft normiert ist.
- Anschließend sind Motor- und Bremskraft abgebildet. Wegen der unterschiedlichen Motorleistung ist die Motorkraft  $F_M$  auf ihr fahrzeug- und streckenabhängiges Maximum  $\frac{P_M}{v(s)}$  gemäß (5.43) normiert. Die Bremskraft  $F_B$  ist auf die maximale Bremskraft  $F_{B,\min} = -\mu_x G$  gemäß (E.3) normiert. Durch die linear skalierte Darstellung der Summe der unterschiedlich normierten Größen ergeben sich Knicke im Übergang von Beschleunigen zu Verzögern, die im tatsächlichen Verlauf von  $F_M + F_B$  nicht vorhanden sind (vgl. Bild 5.5).
- Das unterste Diagramm zeigt die Differenz der Fahrzeit beider Fahrzeuge. Dabei ist zu beachten, dass die Zeitmessung zwar bei der Start-/Zielgeraden bei  $s = 0$  m beginnt, die Fahrzeuge bei der Rundkurs-Optimierung aber quasi einen „fliegenden Start“ machen, d. h. dass sie am Nullpunkt bereits eine gewisse Geschwindigkeitsdifferenz haben. Fzg 2 benötigt für den Rundkurs 112,6 s und Fzg 1 127,2 s.

Durch die Zeitoptimalität ist fast überall mindestens eine UNB aktiv: entweder ist die Motorkraft  $F_M$  oder mindestens eine Reifen-Horizontalkraft am Limit. Lediglich in Übergängen gibt es kurze Bereiche, in denen entweder aufgrund der Fahrdynamik oder durch die sekundären Gütemaße keine UNB aktiv ist, z. B. für Fzg 2 im Übergang von Beschleunigen zu Verzögern bei  $s = 3350 \dots 3450$  m in Bild 5.11.

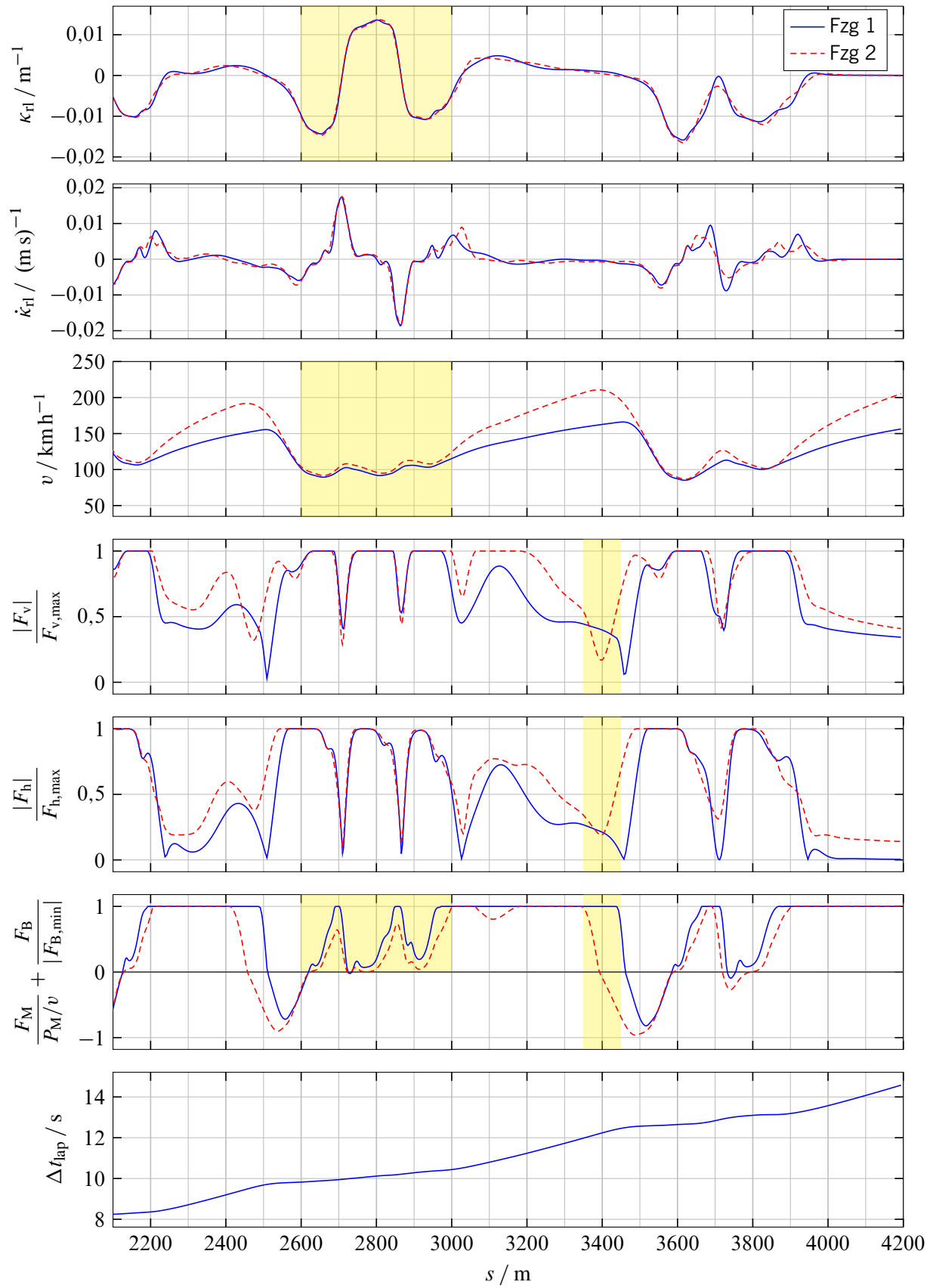
Betrachten wir nun den Apex von Kurve 10 bei  $s = 1570$  m: Erwartungsgemäß ist das Minimum der Geschwindigkeit mit  $v = 53$  km/h exakt im Apex, sodass genau dort auch der Nulldurchgang von  $(F_M + F_B)$  liegt. Die maximale Verzögerung wird 56 m früher bei  $s = 1514$  m erreicht. Weil die Rennlinie dort einen leichten Rechtsbogen ( $\kappa_{rl} < 0$ ) hat, ist keine volle Verzögerung möglich. Nun kann man in Bild 5.10 erkennen, dass die Bremse ab diesem Punkt bis zum Apex kontinuierlich geöffnet wird, während die Krümmung betragsmäßig immer weiter zunimmt. Gleichzeitig sind die Reifenkräfte ab dem Bremspunkt bis zum Kurvenausgang permanent am Limit (zuerst hinten, dann vorne). Das zeigt gerade das gewünschte Verhalten: hart anbremsen vor der Kurve, dann gleichzeitig die Bremse öffnen und so stark einlenken, wie es der Kraftschluss erlaubt, hinter dem Apex dann die Lenkung öffnen und gleichzeitig soviel Gas geben, wie es bei der Zentripetalbeschleunigung möglich ist.

Eine weitere interessante Stelle ist die Kurvenkombination 15 – 17 bei  $s = 2600 \dots 3000$  m: Man kann in Bild 5.11 sehen, dass die Kurvenkombination im Vergleich zu den Bereichen davor und dahinter relativ langsam durchfahren werden muss. Die Kurven folgen so schnell aufeinander, dass dazwischen kaum Beschleunigungsphasen möglich sind. Der interessante Aspekt zeigt sich im



**Bild 5.10:** Verlauf von Krümmung, Geschwindigkeit und normierten Kräften an Reifen, Antrieb und Bremse sowie die Fahrzeitdifferenz von den Ideallinien aus Bild 5.9 für  $s \leq 2100$  m





**Bild 5.11:** Fortsetzung von Bild 5.10 für  $s \geq 2100$  m. Beachte die unterschiedliche Skalierung der Ordinate bei  $\kappa_{rl}$  und  $\dot{\kappa}_{rl}$ .

Verlauf der Ideallinie in Bild 5.9. Üblicherweise werden Kurven außen angefahren, im Apex verläuft die Ideallinie am Innenrand und tangiert am Kurvenausgang wieder den Außenrand. Dieses prinzipielle Schema trifft auf die beiden äußeren Kurven 15 und 17 zu, nicht aber auf die mittlere Kurve 16, wo die Ideallinie durchweg am Innenrand verläuft und daher eine hohe Krümmung mit entsprechend niedriger Geschwindigkeit hat. Der Grund hierfür ist, dass die mittlere Kurve 16 zugunsten der nachfolgenden geopfert wird. Man muss sie ohnehin recht langsam durchfahren, und wenn man sie am Innenrand durchfährt, verschafft man sich eine gute Ausgangsposition für die nächste Kurve, weil man den Apex der nächsten Kurve etwas vorverlegen kann und so früher beschleunigt. Weil hinter Kurve 17 eine lange Beschleunigungspassage anschließt, bringt eine hohe Kurvenausgangsgeschwindigkeit einen großen Vorteil.

Während die fahrdynamischen Größen an vielen Stellen deutliche Unterschiede aufweisen, unterscheiden sich die beiden Ideallinien an den meisten Stellen nur geringfügig. Da man in der Übersichtsdarstellung in Bild 5.9 die Unterschiede zwischen den Ideallinien nur schwer erkennen kann, ist die langgezogene Kurve 4 im Bereich  $s = 600 \dots 1100$  m, wo eine recht große Abweichung von  $\Delta d \approx 3$  m vorliegt, zusätzlich vergrößert abgebildet. Der Grund für diese Abweichung liegt in der unterschiedlichen Motorleistung: Fzg 1 kann auf dem gesamten Abschnitt voll beschleunigen, ohne dass die Kraftschlussgrenze erreicht wird (s. Bild 5.10). Deswegen entsteht die Ideallinie lediglich durch die Minimierung der Weglänge. Anders verhält es sich bei Fzg 2, wo nur bis  $s \approx 880$  m voll beschleunigt werden kann. Danach muss die Motorkraft reduziert werden, weil ansonsten die Vorderachse ausbrechen würde. Es ist wichtig, an der kritischsten Stelle, nämlich dem „Mini-Apex“ bei  $s = 900$  m, eine möglichst hohe Geschwindigkeit halten zu können. Man kann in der Ausschnittvergrößerung der Krümmung  $\kappa_{rl}$  in Bild 5.10 sehen, dass genau dort die Krümmung für Fzg 2 minimiert wird, was durch eine leichte Rechts-Links-Kombination zuvor bei  $s = 700 \dots 800$  m erkaufte wird. Auch in Bild 5.9 kann man den leichten Bogen erkennen. In Zahlen ist die Krümmung für Fzg 2 im Apex um  $\Delta \kappa_{rl} \approx 6 \cdot 10^{-4} \text{ m}^{-1}$  niedriger als bei Fzg 1. Dies führt unter einer quasi-statischen Betrachtung gemäß (2.1) auf S. 14 bei einer Geschwindigkeit von  $v = 190 \text{ km/h}$  zu einer Verringerung der Zentripetalbeschleunigung von

$$\Delta a_c = \Delta \kappa_{rl} \cdot v^2 \approx 6 \cdot 10^{-4} \frac{1}{\text{m}} \cdot \left(53 \frac{\text{m}}{\text{s}}\right)^2 \approx 1,7 \frac{\text{m}}{\text{s}^2},$$

was für eine Erhöhung der Längsbeschleunigung genutzt werden kann.

## 5.6 Erweiterung zur „Komfort-Planung“

Der Fokus liegt in der vorliegenden Arbeit auf zeitoptimalen Trajektorien im fahrdynamischen Grenzbereich. Aber wie der Titel bereits ausdrückt, ist die Planung nicht nur *im* Grenzbereich einsetzbar, sondern mit gewissen Erweiterungen auch im Komfort-Bereich für den realen Straßenverkehr, aber mit dem Potential, *bis in* den Grenzbereich agieren zu können. Welche physikalischen Größen oder Reize das subjektive Komfortempfinden beeinflussen, ist eine komplexe Fragestellung, bei der viele Aspekte der menschlichen Physiologie und Psychologie eine Rolle spielen. Eine quantitative Bewertung von Komfort, lässt sich objektiv kaum definieren, sodass eine allge-

meingültige Angabe von Richt- bzw. Grenzwerten nicht ohne weiteres möglich ist [27]. Im Extremfall kann eine Trajektorie zu Kinetose (Bewegungskrankheit, engl. *motion sickness*) führen, die nach [74] „durch verschiedene Beschleunigungsreize ausgelöst werden kann, die außerhalb der individuellen Adaptationsschwelle liegen“. Die Trajektorie ist dann für die betreffende Person offensichtlich sehr unkomfortabel. Doch soll hier weder das weite Feld der Kinetose behandelt werden noch eine Untersuchung der vielfältigen Einflussparameter für Komfortempfinden durchgeführt werden, sondern eine Methode vorgestellt werden, mit der sich diese Trajektorienplanung auf spezielle Komfortkriterien aus dem Bereich der Fahrdynamik applizieren lässt.

Die wohl naheliegendsten fahrdynamischen Kriterien für den Komfort einer Trajektorie sind ihre Beschleunigungen und Rucke (in alle Richtungen), was u. a. auch in [24, 27, 74] angeführt wird. Auch wenn die Vertikalbewegung sicher Einfluss hat, werden hier nur Horizontalbewegungen betrachtet, um das vorhandene Fzg-Modell verwenden zu können. Der Ruck kann bereits durch die Sekundär-Gütemaße in Abschnitt 5.2.2 reduziert werden, wobei anstelle des Querrucks die Krümmungsänderung verwendet wird, welche den Querruck maßgeblich beeinflusst. Als 2. Ableitung der Geschwindigkeit beschreibt der Ruck die höherfrequente Dynamik. Im Folgenden wird zusätzlich die niederfrequente Dynamik betrachtet, die sich in der Beschleunigung widerspiegelt.

Es dürfte evident sein, dass eine Trajektorie am fahrdynamischen Limit, deren Beschleunigung auf dem Kammschen Kreis liegt, allgemein als nicht komfortabel eingestuft wird. Das gilt erst recht, wenn man das Sicherheitsempfinden als weiteres Komfortkriterium heranzieht, vgl. [27]. Dennoch erfüllt eine zeitoptimale Ideallinie im Grunde bereits ein wesentliches Komfortkriterium, indem sie die Querbeschleunigung minimiert. Zudem verknüpft das zeitoptimale Gütemaß auf elegante Weise die Längs- und Querbewegung, sodass sich ein harmonischer Verlauf ergibt. Nur fühlt sich die Ideallinie wegen der hohen Geschwindigkeit trotzdem nicht komfortabel an. Würde man sie einfach langsamer abfahren, würde sie teilweise unmotiviert oder nicht intuitiv wirken, weil eben diese Verknüpfung von Längs- und Querbewegung für ein niedriges Geschwindigkeitsniveau nicht mehr optimal ist. Daher sind für die Komfort-Planung weitere Kriterien notwendig.

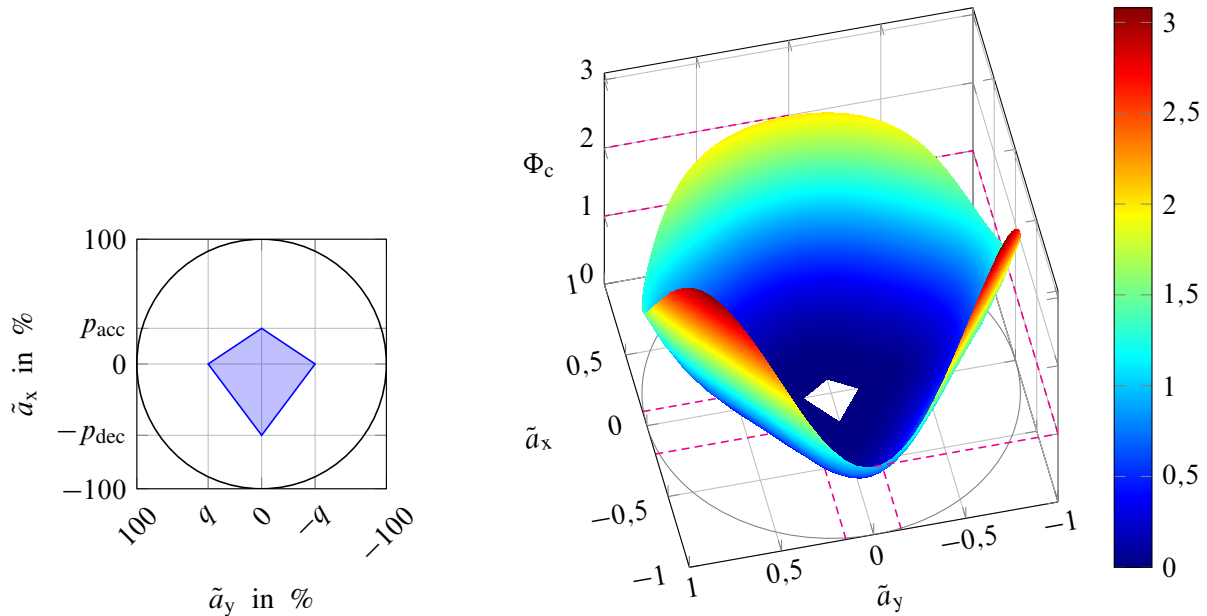
Geht man davon aus, dass die meisten Fahrer selbst so fahren, wie sie es als angenehm oder komfortabel empfinden, kann man aus gemessenen Fahrprofilen Komfortkriterien ableiten. Dabei fällt auf, dass die meisten Fahrer kombinierte Längs- und Querbeschleunigungen vermeiden [70], d. h. der Kammsche Kreis wird entlang der Diagonalen eingedrückt. Hierfür wird im Folgenden ein Potential  $\Phi_c$  für das Komfort-Gütemaß

$$J_c = \sum_{k=1}^N \Phi_c(\mathbf{u}_k, \mathbf{x}_k) \quad (5.45)$$

kreiert.

### 5.6.1 Komfort-Potential

Ob das GG-Diagramm einer „komfortablen“ Trajektorie eher ein konvexer Bereich (d. h. noch kreis-ähnlich oder rautenförmig) oder eher ein konkaver Bereich (d. h. schon „eingedellt“) ist,



(a) Über die Parameter  $p_{acc}$ ,  $p_{dec}$  und  $q$  wird eine Raute im GG-Diagramm definiert, deren Inneres potentialfrei ist.

(b) Potential  $\Phi_c$  exemplarisch mit den Parametern  $p_{acc} = 10$ ,  $p_{dec} = 20$  und  $q = 15$  sowie  $w_{acc} = 2$  und  $w_{dec} = 1$ . Die Punkte in der Mitte mit  $\Phi_c = 0$  sind transparent dargestellt.

**Bild 5.12:** Das Komfort-Potential wird zum einen über die Eckpunkte der potentialfreien Fläche bei niedriger Beschleunigung und zum anderen über das Maximum bei reiner Längsbeschleunigung parametrisiert.

dürfte sehr subjektiv sein. Für die Optimierung ist es aber von Vorteil, wenn das Potential konvex ist. Darum definiert hier eine Raute – oder genauer gesagt ein Drachenviereck – das Gebiet im GG-Diagramm, innerhalb dessen die Trajektorienplanung frei planen kann, d. h. kein zusätzliches Potential vorliegt. Es ist in Bild 5.12(a) blau hinterlegt und durch die 3 Parameter  $p_{acc}$ ,  $p_{dec}$  und  $q$  definiert, welche in Prozent der maximalen Beschleunigung angegeben sind.

Das Äußere der Raute kann als implizite Gleichung in Abhängigkeit der relativen Beschleunigungen  $\tilde{a}_{x/y}$  durch

$$C_{acc,li}(\tilde{a}_x, \tilde{a}_y) + C_{acc,re}(\tilde{a}_x, \tilde{a}_y) + C_{dec,li}(\tilde{a}_x, \tilde{a}_y) + C_{dec,re}(\tilde{a}_x, \tilde{a}_y) > 0 \quad (5.46)$$

beschrieben werden, worin die 4 Ränder durch  $C_{...}(\tilde{a}_x, \tilde{a}_y)$  mit  $p_{acc}$ ,  $p_{dec}$ ,  $q \in (0, 100)$  als

$$C_{acc,li/re}(\tilde{a}_x, \tilde{a}_y) := \max \left\{ 0, \frac{\tilde{a}_x}{10^{-2} p_{acc}} \pm \frac{\tilde{a}_y}{10^{-2} q} - 1 \right\} \quad (5.47a)$$

$$C_{dec,li/re}(\tilde{a}_x, \tilde{a}_y) := \max \left\{ 0, -\frac{\tilde{a}_x}{10^{-2} p_{dec}} \pm \frac{\tilde{a}_y}{10^{-2} q} - 1 \right\} \quad (5.47b)$$

definiert sind. Bei der exakten Berechnung der Beschleunigungen gehen fast alle Modellvariablen ein. Da es sich bei dieser Funktion aber nur um ein unkritisches Zusatzpotential handelt, sollte ein sinnvoller Kompromiss zwischen Detaillierungsgrad und Rechenzeit bzw. Komplexität gefunden werden: Wie in (3.50) auf S. 42 wird beim Kräftegleichgewicht der Lenkwinkel zu

$\delta \equiv 0$  angenommen, damit  $\tilde{a}_x$  nur von Reifenlängskräften und  $\tilde{a}_y$  nur von Reifenquerkräften abhängt. Natürlich geht in die Querkraft selbst der echte Lenkwinkel ein. In Längsrichtung wird der Luftwiderstand vernachlässigt, um die Geschwindigkeit zu eliminieren. Außerdem wird der Einfluss der dynamischen Achslastverteilung auf die Maximal-Beschleunigung vernachlässigt, da die Achslastverteilung im Komfortbereich weniger relevant ist. Es folgt mit  $F_{y,r,rel} := \frac{1}{D} F_{y_r}$  aus (3.23):

$$\tilde{a}_x := \frac{F_M + F_B}{\mu_x m g}, \quad \tilde{a}_y := \frac{m_v}{m} F_{y_r,v,rel} + \frac{m_h}{m} F_{y_r,h,rel} \quad (5.48)$$

Das Potential  $\Phi_c$  soll in radialer Richtung monoton steigen und zudem zweimal stetig differenzierbar sein. Es wird daher aus kubischen Funktionen gebildet:

$$\Phi_c = \frac{1}{3} \left( \lambda_{\text{comf,acc}} (C_{\text{acc,li}}^3 + C_{\text{acc,re}}^3) + \lambda_{\text{comf,dec}} (C_{\text{dec,li}}^3 + C_{\text{dec,re}}^3) \right) \quad (5.49)$$

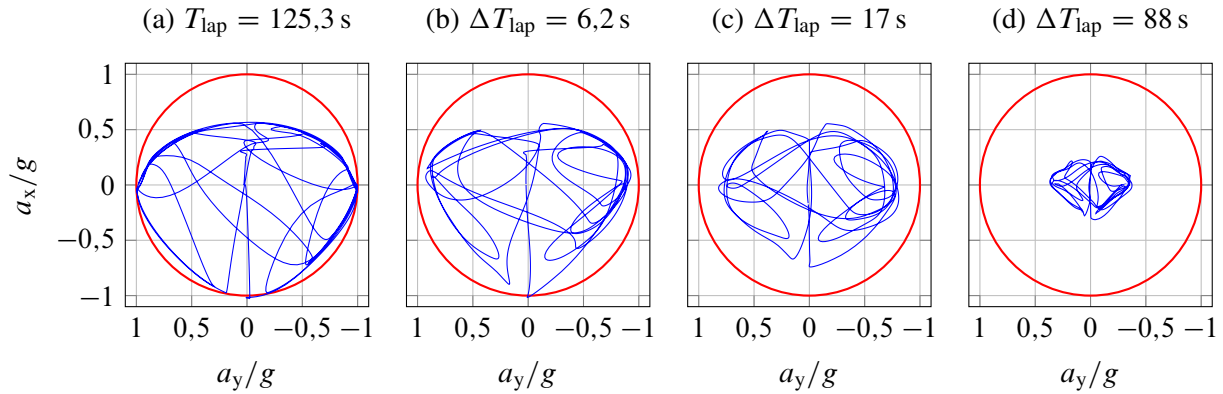
Die Parameter  $\lambda_{\text{comf}}$  ergeben sich, indem  $\Phi_c(\tilde{a}_x, \tilde{a}_y)$  für  $\tilde{a}_y = 0$  an den beiden Stellen  $\tilde{a}_x = \pm 1$  vorgegeben wird, nämlich  $\Phi_c(+1, 0) \stackrel{!}{=} w_{\text{acc}} \geq 0$  und  $\Phi_c(-1, 0) \stackrel{!}{=} w_{\text{dec}} \geq 0$ , siehe Bild 5.12(b). Es folgt

$$\lambda_{\text{comf,acc}} = \frac{3}{2} \left( \frac{p_{\text{acc}}}{100 - p_{\text{acc}}} \right)^3 w_{\text{acc}} \quad \text{und} \quad \lambda_{\text{comf,dec}} = \frac{3}{2} \left( \frac{p_{\text{dec}}}{100 - p_{\text{dec}}} \right)^3 w_{\text{dec}}. \quad (5.50)$$

## 5.6.2 Ergebnisse

Das Komfort-Gütemaß bietet in Kombination mit den Gütemaßen aus Abschnitt 5.2.2 die Möglichkeit, der Trajektorie verschiedenste Charakteristiken zu geben und trotzdem dem Anspruch gerecht zu werden, nötigenfalls bis ans fahrdynamische Limit zu agieren. Hauptkriterium ist dabei, dass die Trajektorie in sich stimmig ist, d. h. die Art, wie vor einer Kurve gebremst, wie sie durchfahren und wie herausbeschleunigt wird, muss harmonisch wirken, und wenn der Passagier die Charakteristik einmal kennengelernt hat, sollte das Verhalten während der weiteren Fahrt auf ihn vorhersehbar wirken. Die Komfortkriterien stehen im Kompromiss zur Fahrzeit, die es weiterhin möglichst gering zu halten gilt, schließlich ist auch im Alltag meist nicht der Weg das Ziel. Dieser Kompromiss muss dann für spezielle Vorgaben appliziert werden. Vorteilhaft ist, dass die Struktur des Optimierungsproblems und erst recht der Planer selbst dabei unverändert bleiben. „Wollen die Ingenieure dann die Charakteristik ändern, müssen sie im Grunde nur die Vorgabeparameter anpassen“, schreibt eine Redakteurin von *auto motor und sport* bei einer Demo-Fahrt auf dem *Autódromo do Algarve* [8], „Doch genau dafür bedarf es immenser Erfahrung, um einschätzen zu können, welcher Parameter welche Auswirkungen hat. [...] Wie viel Aufwand das ist, lässt sich nur errahnen, das Ergebnis ist aber deutlich spürbar. Während Walter, der für die Tests genutzte Audi RS7, uns in der einen Runde teils noch kräftig in die Sitze drückt, spät in die Kurven hineinbremst und ab dem Scheitelpunkt wieder ordentlich das Tempo anzieht, kutschiert er uns wenige Klicks später viel sanfter und ausgeglichener über den Kurs.“

Bild 5.13 zeigt GG-Diagramme und die daraus resultierenden Rundenzeiten für unterschiedliche Parametrierungen bei  $\mu_x = \mu_y = 1$ . In (a) ist  $w_{\text{acc}} = w_{\text{dec}} = 0$ . Dass die äußeren Punkte beim



**Bild 5.13:** Je nach Parametrierung ergeben sich unterschiedliche GG-Diagramme im Kompromiss zur Rundenzeit.

Verzögern nicht auf dem Kammschen Kreis bei  $1g$  liegen, kommt daher, dass durch die dynamische Achslastverteilung (Abschnitt 3.2.3) in Kombination mit der degressiven Reifenkennlinie (Bild 3.4 auf S. 33) die Vorderachse weniger zusätzliche Kraft aufbauen kann, als die Hinterachse verliert. In Folge beträgt die maximal mögliche Verzögerung im SP je nach Fahrsituation weniger als  $1g$ . Hingegen kann beim reinen Verzögern aus hoher Geschwindigkeit aufgrund des Luftwiderstands eine Verzögerung  $z_B > 1$  erreicht werden. Wie sich die Achslastverteilung speziell auf die SP-Beschleunigung auswirkt, wird bspw. in [79] untersucht.

Während bei den Komfort-Planungen (b) und (d) die spitze Rautenform deutlich erkennbar ist, ist (c) runder, was über die Steigung von  $\Phi_c$  beeinflusst werden kann. Bei (b) und (d) sind die lateralen Maxima leicht nach oben verschoben. Dies bewirkt ein früheres Beschleunigen aus der Kurve, d. h. ganz kurz vor dem Apex wird schon mit einer leichten Beschleunigung begonnen, was sich gerade bei langsamer Fahrt natürlicher und agiler anfühlt. Erreicht wird dieser Effekt durch eine asymmetrische Parametrierung in Längsrichtung, sodass das Minimum von  $\Phi_c$  in positive  $a_x$ -Richtung verschoben ist.

Zwar sind für eine vollständige Abstimmung der Trajektorienplanung auf bestimmte Fahrprofile sicher noch einige zusätzliche Güteterme oder Anpassungen notwendig, jedoch ermöglichen bereits die vorhandenen Applikationsmöglichkeiten die Einstellung der grundlegenden Charakteristik. Auch wenn die exakte Auswirkung einer Parametrierung ohne Simulation oft schwer abschätzbar ist, sind alle Parameter so gewählt, dass sie physikalisch interpretierbar sind: Damit z. B. die Planung in *einem* Punkt ein um  $\Delta\Phi_c$  höheres Potential annimmt, muss der Zeitgewinn auf der *gesamten* Strecke ebenfalls bei  $\Delta\Phi_c$  liegen. So ergeben sich relativ niedrige Gewichtungen, wenn die Planung von selbst bis nahe an den Rand gehen soll; bspw. ist in Bild 5.13(b)  $w_{acc} = w_{dec} = 1,9 \cdot 10^{-4}$  und in Bild 5.13(d)  $w_{acc} = w_{dec} = 2$ . Dennoch wird die Planung in einer Notsituation immer ans Limit gehen, was in Abschnitt 7.4.2 gezeigt wird.

## 6 Toolchain der Trajektorienplanung

Die Trajektorienplanung benötigt eine Referenzlinie (vgl. *Definition der Referenzlinie* auf S. 7), auf deren Bogenlänge  $s$  sich alle übrigen Größen beziehen und deren Krümmung  $\kappa$  den Streckenverlauf angibt. Wurde eine Referenzlinie einmal erstellt, kann sie in Kartendaten hinterlegt werden und muss daher nicht in Echtzeit berechnet werden. Abschnitt 6.1 beschreibt, wie sie aus den Fahrbahnrandern berechnet wird, welche in unserem Projekt in Form kartesischer Punkte mittels DGPS gemessen wurden, und durch Messrauschen gewisse Artefakte aufweisen können.

Die Referenzlinie verläuft in der Nähe der Mittellinie, welche aber i. A. nicht stetig differenzierbar ist, sodass ihre Krümmung verrauscht ist. Aus zwei Gründen leidet darunter die Genauigkeit der Planung: Die Diskretisierung der Bahnplanung geht davon aus, dass die Krümmung zwischen zwei aufeinanderfolgenden Abtastschritten konstant ist, was bei einem verrauschten Krümmungsverlauf nicht mehr zutrifft. Des weiteren wird die Bahn innerhalb der Planung durch den Querversatz zur Referenz beschrieben, welcher exakt orthogonal zur Referenz ist. Am ausgangsseitigen Interface der Planung wird der Querversatz gemäß (3.11) auf S. 28 in kartesische Koordinaten rücktransformiert. Bei einer nicht stetig differenzierbaren Referenz ist das Lot an der Referenz, also der Normalenvektor in (3.11), in seiner Orientierung unstetig. Daher können bei der Rücktransformation Artefakte oder sogar „Knicke“ im Linienverlauf entstehen, die im krummlinigen Koordinatensystem der Planung nicht vorhanden sind. Deswegen wird die Referenz geglättet, wofür ein eigener Algorithmus entwickelt wurde, welcher in Abschnitt 6.2 erklärt wird.

Kennt man die  $s$ -Koordinate und den Querversatz  $d$  eines Punktes, lässt er sich über (3.11) leicht in ein kartesisches Koordinatensystem transformieren. Schwieriger ist der umgekehrte Fall, in welchem man die kartesischen Koordinaten eines beliebigen Punkts kennt und wissen möchte, bei welchem  $s$ -Wert auf der Referenz das Lot genau durch diesen Punkt verläuft. Dieser Fall tritt z. B. dann auf, wenn man die Position des Ego-Fzgs ermittelt hat und die dazugehörige  $s$ -Koordinate auf der Referenz benötigt. Die Berechnung hierfür, die im Folgenden als *Matching* bezeichnet wird, ist in Abschnitt 6.3 beschrieben.

Fast alle Berechnungen im aktuellen Kapitel sind geometrischer Natur. Folgende Festlegungen werden dafür vorgenommen:

- Punkte (d. h. geometrische Raumpunkte) werden mit Großbuchstaben bezeichnet und in kartesischen Koordinaten angegeben.
- Ortsvektoren zu den Punkten werden durch die entsprechenden Kleinbuchstaben mit Vektorpfeil angegeben. Der Vektorpfeil anstelle der bisher verwendeten fett gedruckten Symbole zeigt, dass es sich um einen Vektor aus unserem Anschauungsraum handelt.
- Die Fahrbahn wird lokal als eben angenommen, sodass die Fahrbahnoberfläche in einer Umgebung eines Punkts auf der Referenz über die gesamte Fahrbahnbreite hinweg als eine

zweidimensionale Ebene beschrieben werden kann. Das globale kartesische Koordinatensystem  $(x_0, y_0, z_0)$  ist ortsfest (Ursprung beliebig) und um die lokalen Fahrbahnneigungswinkel  $\theta(s)$  und  $\phi(s)$  gedreht, sodass seine  $x_0y_0$ -Ebene in der Fahrbahnebene liegt und  $z_0$  überall orthogonal zur Fahrbahn steht. Durch diese zweidimensionale Mannigfaltigkeit können Referenz- und Randlinien vollständig im  $\mathbb{R}^2$  beschrieben werden.

- Um dennoch das Vektorprodukt zweier Vektoren  $\vec{p}, \vec{q} \in \mathbb{R}^2$  auswerten zu können, wird die 3. Koordinate implizit zu null ergänzt, wodurch sich ein Vektor in  $z_0$ -Richtung ergibt:

$$\vec{p} \times \vec{q} = \begin{bmatrix} p_x \\ p_y \\ 0 \end{bmatrix} \times \begin{bmatrix} q_x \\ q_y \\ 0 \end{bmatrix} = (p_x q_y - q_x p_y) \vec{e}_3 \quad \text{mit} \quad \vec{e}_3 := \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (6.1)$$

- $\overline{PQ}$  bezeichnet die Verbindungslinie zwischen  $P$  und  $Q$ , und  $|\overline{PQ}| = |\vec{q} - \vec{p}|$  ihre Länge.

## 6.1 Erstellung der Referenzlinie

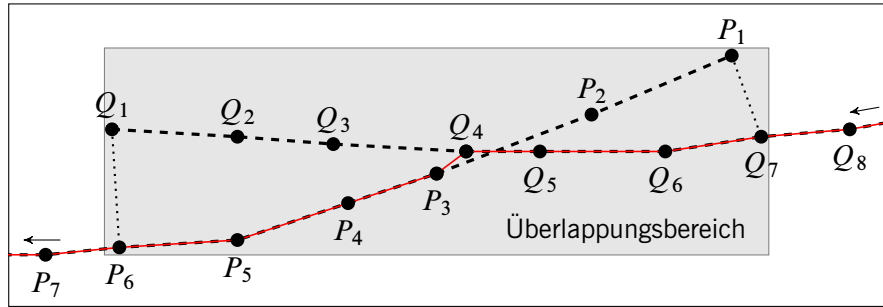
Aus den diskreten Punkten der Ränder sollen zuerst diskrete Punkte der Mittellinie berechnet werden, welche als Stützstellen einer anschließenden Spline-Interpolation dienen. Aus der Mittellinie wird dann durch Filterung und Abtastung die Referenzlinie gewonnen, welche nach Vorgabe im Projekt maximal 30 % der Fahrbahnbreite von der Mittellinie abweichen sollte. Die Algorithmen sollen so robust sein, dass sie bei unterschiedlicher Beschaffenheit und Qualität der Randpunkte bestmögliche Ergebnisse liefern. Dafür sind viele Subroutinen nötig, die potentielle Probleme beheben können. Die wichtigsten sind im Folgenden erklärt.

### 6.1.1 Preprocessing der Fahrbahnränder

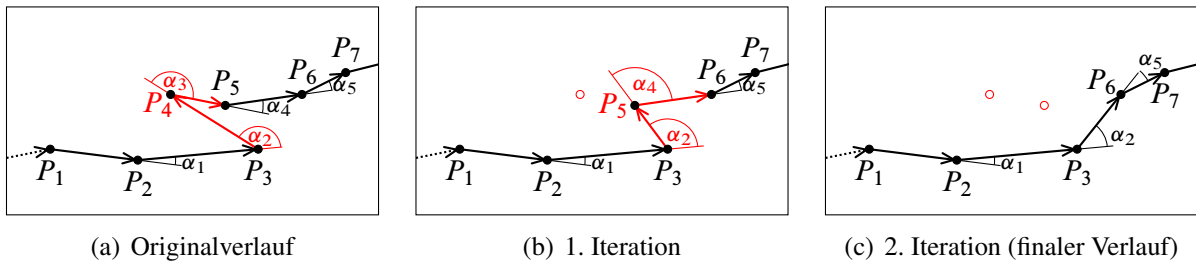
Der erste Schritt ist eine Vorverarbeitung der Randdaten mit dem Ziel, aus den gemessenen Punkten eine monotone Linie zu generieren, weil die Algorithmen voraussetzen, jedem Punkt einer Referenzlinie höchstens ein Segment auf der Randlinie zuordnen zu können. Hierfür werden zuerst redundante oder beinahe redundante Punkte zu einem Punkt zusammengefasst.

Bei Rundkursen werden Anfang und Ende der Messdaten gesondert behandelt. Bild 6.1 zeigt einen typischen Fall, in welchem sich Anfang und Ende überlappen, jedoch nicht ganz parallel sind, sodass sich die Linie im Übergang schneidet. Die Menge aller Punkte wird in der Hälfte in zwei disjunkte Teilmengen unterteilt. Die Punkte der ersten Menge sind in Bild 6.1 fortlaufend mit  $P_1, P_2, \dots$  benannt, die Punkte der letzten vom Ende her mit  $Q_1, Q_2, \dots$ . Der Übergang soll bei dem dichtesten Punktpaar  $(P_i, Q_j)$  innerhalb des Überlappungsbereichs sein. Da sich der Suchraum mühelos auf diesen relativ kleinen Bereich eingrenzen lässt, kann zur Bestimmung dieses Punktpaares ein naiver Brute-force-Algorithmus anstelle eines rekursiven wie in [21, Kap. 33.4] zum Einsatz kommen. Der Überlappungsbereich ergibt sich wie in Bild 6.1 dargestellt durch ein





**Bild 6.1:** Anfang und Ende einer Randlinie eines Rundkurses schneiden sich in einem Überlappungsbereich. Sie werden an ihrem dichtesten Punktpaar verbunden, sodass sich eine geschlossene Linie (rot durchgezogen) ergibt.



**Bild 6.2:** Rücksprünge im Linienverlauf werden anhand der Winkel zwischen den Segmenten erkannt und iterativ entfernt.

Rechteck, das die Punkte  $P_1$  und  $Q_1$  sowie ihre beiden dichtesten Punkte aus der jeweils anderen Teilmenge (hier  $P_6$  und  $Q_7$ ) einschließt. Der Übergang erfolgt in Bild 6.1 bei  $(P_3, Q_4)$ .

Die Randpunkte liegen größtenteils in sortierter Reihenfolge vor, jedoch kann es passieren, dass durch Messrauschen Rücksprünge entstehen, siehe Bild 6.2(a). Ein solcher Rücksprung liegt vor, wenn die Winkel  $\alpha_i$  und  $\alpha_{i+1}$  zwischen 2 aufeinanderfolgenden Liniensegmenten beide größer als  $90^\circ$  sind. Die Winkel ergeben sich aus dem Skalarprodukt der normierten Richtungsvektoren:

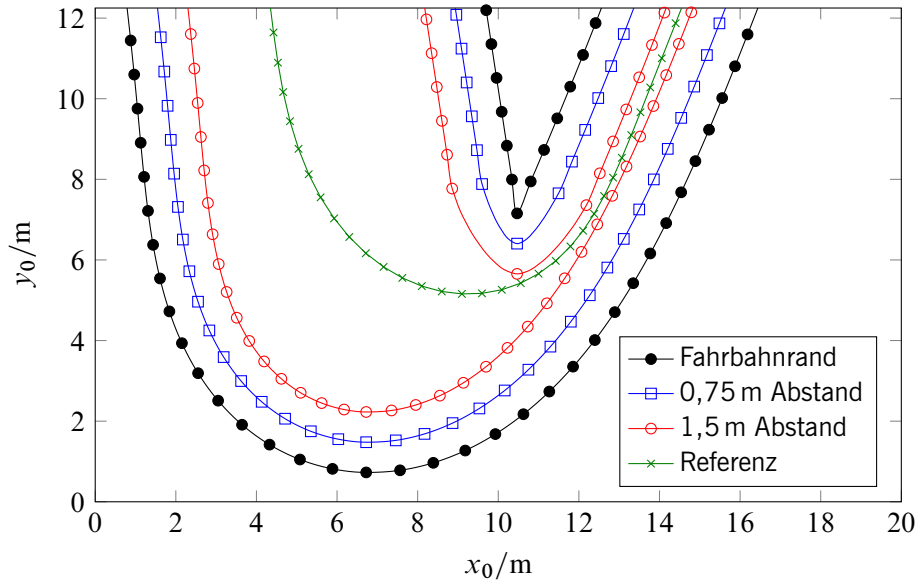
$$\cos(\alpha_i) = \frac{\vec{r}_i \cdot \vec{r}_{i+1}}{|\vec{r}_i| |\vec{r}_{i+1}|}, \quad \vec{r}_i := \vec{p}_{i+1} - \vec{p}_i, \quad i = 1, \dots, N-1 \quad (6.2a)$$

$$= \vec{r}_{n,i} \cdot \vec{r}_{n,i+1}, \quad \vec{r}_{n,i} := \frac{\vec{r}_i}{|\vec{r}_i|} \quad (6.2b)$$

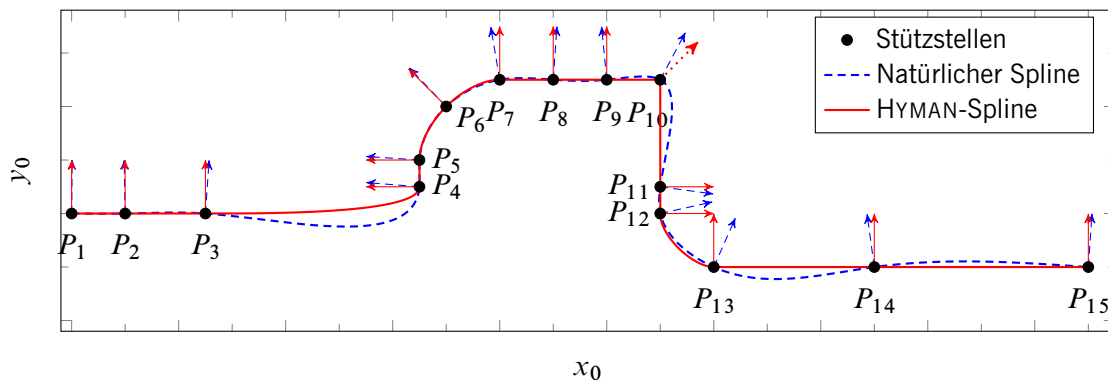
Wurde ein Rücksprung erkannt, wird der entsprechende Punkt, wie in Bild 6.2(b) abgebildet, entfernt. Das Prozedere wird wiederholt, bis keine Rücksprünge mehr vorhanden sind, s. Bild 6.2(c).

### 6.1.2 Sicherheitsabstand

Die Grenzen  $d_{re/li}$  in (5.35e) auf S. 81 sind so definiert, dass die halbe Fzg-Breite abgezogen ist, sodass die Fahrlinie – also die Bahn des Schwerpunkts – diese Grenze tangiert, wenn das äußere Rad den Fahrbahnrand tangiert. Genauso wurde der Fahrbahnrand (im Folgenden einfach „Rand“ genannt) auch in allen bisherigen Graphiken abgebildet. Zusätzlich kann ein Sicherheitsabstand



**Bild 6.3:** Die Parallelverschiebung des Fahrbahnrandes liefert auch bei großen Sicherheitsabständen und spitzen Winkeln im Fahrbahnrand saubere Linien mit konstantem Abstand.



**Bild 6.4:** Die Interpolation der Stützstellen mittels natürlichem kubischem Spline führt zu unplausiblen Oszillationen über mehrere Stützstellen hinweg. Die kubische Interpolation nach HYMAN ist hingegen co-monoton und liefert einen plausiblen Verlauf. Die Pfeile kennzeichnen jeweils die Normalenvektoren.

zum Rand definiert werden, indem  $d_{\text{re/li}}$  um einen gewissen Wert reduziert wird. Da die Fahrbahnbreite orthogonal zur Referenz angegeben wird und der Sicherheitsabstand orthogonal zum Rand, muss dabei der Winkel zwischen Referenz und Rand einkalkuliert werden. Wenn der Winkel groß ist und sich die Orientierung stark ändert, wie es bei dem Streckenabschnitt in Bild 6.3 der Fall ist, kann es hierbei zu Ungenauigkeiten kommen. Ein zuverlässigeres Verhalten erzielt man, wenn man den Rand parallel verschiebt. Hierzu wird jeder Randpunkt entlang seines Normalenvektors verschoben. Um den Normalenvektor zu bestimmen, wird der Rand mit einem zweidimensionalen kubischen Spline interpoliert, der über die Bogenlänge parametrisiert ist. So kann der Normalenvektor durch Differentiation des Splines gemäß (3.12) auf S. 29 bestimmt werden.

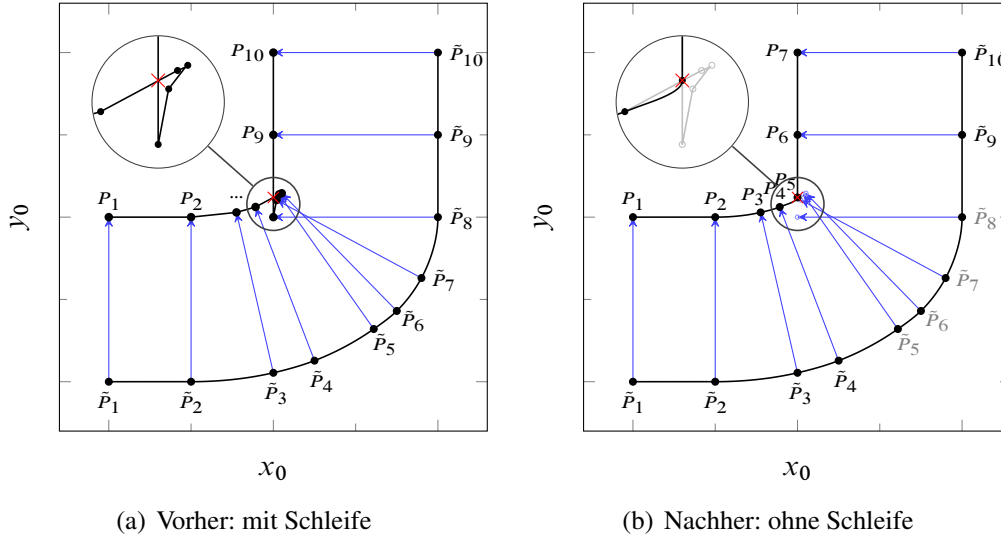
Bei einer Interpolation von Datenpunkten gibt es kein „richtig“ und „falsch“, weil ein korrekter Verlauf zwischen den Punkten nicht definiert ist. Dennoch sollte man beurteilen, wie *plausibel* der Verlauf der gewählten Interpolationsmethode ist. Ein natürlicher Spline ist aufgrund seiner Eigenschaft, zweimal stetig differenzierbare Übergänge an den Stützstellen zu generieren, für diese Anwendung schlecht geeignet. Denn die Folge ist oftmals ein Überspringen, wodurch eine unplausible Interpolation entsteht. Dies zeigt Bild 6.4 (blau gestrichelte Linie) besonders deutlich bei den Segmenten  $P_3 \rightarrow P_4$ ,  $P_9 \rightarrow P_{10} \rightarrow P_{11}$  und  $P_{13} \rightarrow P_{15}$ . Hingegen liefert die kubische Interpolation von HYMAN [42] einen plausiblen Verlauf. Dabei handelt es sich um einen *einmal* stetig differenzierbaren kubischen Spline, mit der besonderen Eigenschaft, dass er co-monoton ist, d. h. lokal monotone Daten werden durch einen lokal monotonen Verlauf interpoliert. Daher bleiben sowohl Monotonie als auch Konvexität der Ausgangsdaten erhalten und es entsteht eine Linie, die näherungsweise der eines menschlichen Zeichners entspricht [42]. Die Co-Monotonie führt dazu, dass die Ableitung innerhalb eines Interpolationsintervalls ihr Vorzeichen nicht ändert, und der Spline niemals überspringt, wie in Bild 6.4 (rot durchgezogenen Linie) zu sehen ist. Somit ist der HYMAN-Spline noch besser geeignet als ein AKIMA-Spline, welcher lediglich Oszillationen über mehrere Segmente hinweg verhindert, jedoch innerhalb eines Segments schwingen kann (im Gegensatz zum natürlichen Spline, der ab  $P_{13}$  über mehrere Segmente oszilliert).

Die genannten Eigenschaften des HYMAN-Splines haben zur Folge, dass 3 oder mehr Punkte, die auf einer Geraden liegen, auch durch eine Gerade interpoliert werden – unabhängig davon wie der weitere Verlauf aussieht. Daher sind in Bild 6.4 die Normalenvektoren in solchen Abschnitten parallel. Allerdings muss man bei der Berechnung der Tangenten- bzw. Normalenvektoren mittels Differentiation beachten, dass man an einer Stützstelle den Nullvektor erhält, wenn das eine angrenzende Segment eine Gerade parallel zu einer der Achsen ist (die Ableitung einer Koordinate ist dann null) und das andere angrenzende Segment im rechten oder spitzen Winkel dazu verläuft, wie es bei  $P_{10}$  in Bild 6.4 der Fall ist. Denn dann haben die Ableitungen beider Koordinaten genau in der Stützstelle ihre Nullstelle. (Die stetige Differenzierbarkeit ist weiterhin gegeben.) Diese Situation tritt i. d. R. nur bei synthetisch generierten Streckenverläufen auf und stellt durchaus keinen Nachteil dar, weil ein „Ersatznormalenvektor“ problemlos durch Mitteln der angrenzenden Vektoren gebildet werden kann (s. gepunkteten Vektor in Bild 6.4).

Die Koeffizienten des HYMAN-Splines können mit der INTEL MKL [44] berechnet werden<sup>1</sup>. Genauso wie der natürliche Spline kann der HYMAN-Spline auch periodisch erstellt werden, d. h. dass Randbedingungen am Anfang und Ende identisch, aber nicht explizit festgelegt sind. Dies ist bei Rundkursen wichtig.

Verschiebt man eine Kurve in Richtung der Kurveninnenseite, überkreuzen sich die Normalenvektoren und es entsteht eine Schleife, wenn die Verschiebung größer als der Radius  $|r| = \frac{1}{|k|}$  ist. Diese Problematik kennen wir schon von der Weg-/Zeit-Transformation in Abschnitt 3.1, jedoch ist sie hier noch wichtiger, weil in den Messdaten lokal sehr hohe Krümmungen auftreten können. Bild 6.5(a) zeigt ein Beispiel für die Entstehung einer Schleife. Um Schleifen in einer Linie zu erkennen, wird jedes Liniensegment auf einen Schnittpunkt mit allen nachfolgenden Segmenten

<sup>1</sup>In der Version 2017 wurde der *Hyman monotonic cubic spline* zur *Data Fitting*-Library der MKL hinzugefügt. In der Version 2019 ist er zwar verfügbar, aber leider nicht mehr in der *Developer Reference* aufgeführt.



**Bild 6.5:** Schleifen im Linienvorlauf, die durch eine Parallelverschiebung einer Linie entstehen können, werden durch den Schnittpunkt zweier Segmente detektiert und entfernt.

überprüft, d. h. es werden die Schnittpunkte aller Geradenpaare

$$g_i : \vec{x} = \vec{p}_i + \lambda (\vec{p}_{i+1} - \vec{p}_i) \quad \text{für } i = 1, \dots, N-2 \quad \text{und} \quad (6.3a)$$

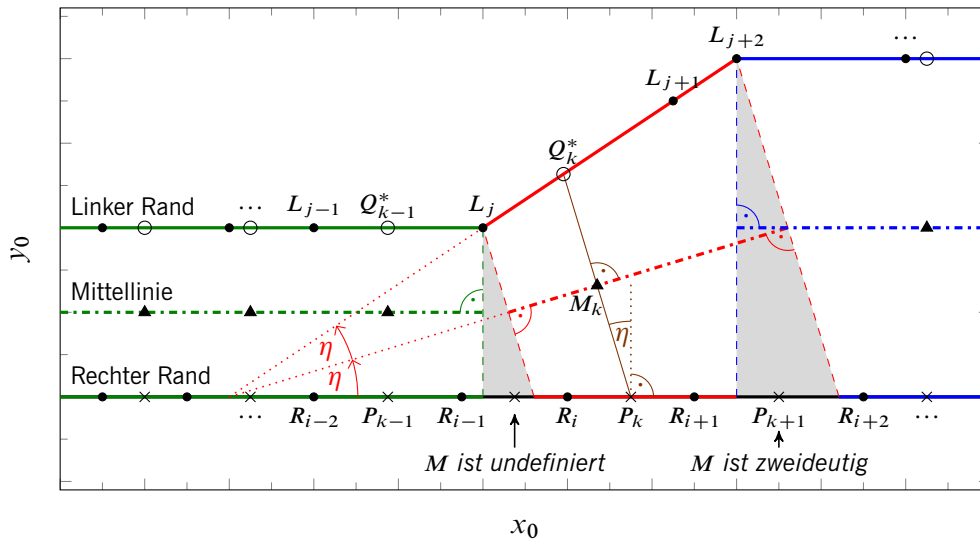
$$h_j : \vec{x} = \vec{p}_j + \rho (\vec{p}_{j+1} - \vec{p}_j) \quad \text{für } j = i+2, \dots, N \quad (6.3b)$$

mit  $\lambda, \rho \in \mathbb{R}$  ermittelt. Ergibt sich für ein Paar sowohl  $\lambda_i \in [0, 1)$  als auch  $\rho_j \in [0, 1)$ , schneiden sich die Segmente, wie es in Bild 6.5(a) für  $i = 4$  und  $j = 8$  dargestellt ist, und der entsprechende Abschnitt ( $P_5 \rightarrow P_8$ ) wird entfernt. Um den Streckenverlauf möglichst wenig zu verfälschen, wird ein Ersatzpunkt am Schnittpunkt eingefügt (s. rotes Kreuz  $P_5$  in Bild 6.5(b)). Sollte der Streckenverlauf eine gewollte Überkreuzung aufweisen, ähnlich wie es bei einem *Figure-8-racing* auftritt, ist dieser Schnittpunkt ausgenommen. Auf den ersten Blick sieht die Auswertung recht aufwendig aus, schließlich müssen  $\sum_{i=2}^{N-2} (N-i) = \frac{1}{2}N(N-3) \lesssim \frac{1}{2}N^2$  Schnittpunkte bestimmt werden. Doch eine analytische Lösung von (6.3) führt zu der relativ einfachen Formel

$$\lambda = \frac{\vec{e}_3 \cdot \left( (\vec{p}_j - \vec{p}_i) \times (\vec{p}_{j+1} - \vec{p}_j) \right)}{\vec{e}_3 \cdot \left( (\vec{p}_{i+1} - \vec{p}_i) \times (\vec{p}_{j+1} - \vec{p}_j) \right)} = \frac{\vec{e}_3 \cdot (\vec{d}_{i,j} \times \vec{r}_j)}{\vec{e}_3 \cdot (\vec{r}_i \times \vec{r}_j)}, \quad (6.4a)$$

$$\rho = \frac{\vec{e}_3 \cdot \left( (\vec{p}_j - \vec{p}_i) \times (\vec{p}_{i+1} - \vec{p}_i) \right)}{\vec{e}_3 \cdot \left( (\vec{p}_{i+1} - \vec{p}_i) \times (\vec{p}_{j+1} - \vec{p}_j) \right)} = \frac{\vec{e}_3 \cdot (\vec{d}_{i,j} \times \vec{r}_i)}{\vec{e}_3 \cdot (\vec{r}_i \times \vec{r}_j)}, \quad (6.4b)$$

worin  $\vec{d}_{i,j} := (\vec{p}_j - \vec{p}_i)$  der Verbindungsvektor der Aufpunkte ist und  $\vec{r}_k := (\vec{p}_{k+1} - \vec{p}_k)$  der Richtungsvektor am Punkt  $P_k$ . Der Nenner enthält das Kreuzprodukt der beiden Richtungsvektoren. Die Gleichungen für  $\lambda$  und  $\rho$  unterscheiden sich nur durch den Index von  $\vec{r}$  im Zähler, wobei  $\vec{r}$  einmal für die gesamte Strecke berechnet werden kann und nicht für jedes Geradenpaar neu berechnet werden muss, sodass jeder Schnittpunkt nur wenige Rechenoperationen erfordert, die zudem gut vektorisiert werden können.



**Bild 6.6:** Die Mittellinie wird über die Winkelhalbierende der abschnittsweise geraden Fahr-  
bahränder berechnet, um lokal denselben orthogonalen Abstand zu den Rändern zu haben.

### 6.1.3 Berechnung der Mittellinie

Die Mittellinie sei als diejenige Linie definiert, die *entlang ihres Lots* denselben Abstand zu den beiden Rändern aufweist. Um an einem Punkt  $M_k$  das Lot fällen zu können, muss eine Annahme für den Verlauf zwischen den Punkten getroffen werden. Hier wird zu Beginn ein abschnittsweise gerader Verlauf angenommen. Im Gegensatz zur Mittellinie verläuft die Referenz i. A. nicht exakt in der Mitte. Zur Berechnung der Referenz wird die Mittellinie mit dem *Kappa-Filter* aus Kapitel 6.2 geglättet und neu abgetastet, sodass die hier berechneten Punkte lediglich als Ausgangspunkte zur Berechnung der Referenz dienen. Während die Referenz zur Berechnung ihrer Krümmung mindestens zweimal stetig differenzierbar und daher möglichst glatt sein muss, kann die Mittellinie diese Bedingung i. A. nicht erfüllen, wie ein einfaches Beispiel zeigt:

Wir betrachten in Bild 6.6 zuerst nur den Verlauf von dem linken und rechten Rand sowie die Mittellinie, ohne die eingetragenen Punkte zu berücksichtigen. Die Strecke lässt sich in 3 Segmente aufteilen, die farblich gekennzeichnet sind: Im 1. und 3. Segment sind die Ränder parallel, sodass der Verlauf der Mittellinie trivial ist. Im 2. Segment ergibt sie sich durch die Winkelhalbierende zwischen den Rändern (s. Winkel  $\eta$ ). Problematisch sind die Übergänge, denn zwischen dem 1. und 2. Segment lässt sich keine Linie angeben, die der Definition der Mittellinie entspricht. Zwischen dem 2. und 3. Segment gibt es gleich zwei Linien, welche die Definition erfüllen.

Folgende Punkte ergeben sich als Fazit dieser Beobachtung:

- Die Definition der Mittellinie kann nur als hinreichende, nicht aber als notwendige Bedingung angesehen werden.
- Auch wenn die Ränder stetig sind, gibt es bei asymmetrischen Rändern Bereiche, in denen es unmöglich ist, eine stetige oder gar stetig differenzierbare Mittellinie anzugeben.

- Darüber hinaus kann es in Haarnadelkurven wie in Bild 6.3 passieren, dass sich für ein Segment des Außenrands (z. B.  $x_0 = 6 \dots 8$  m) kein gegenüberliegendes Segment des Innenrands finden lässt, das für eine sinnvolle Mittenberechnung eingesetzt werden kann.
- Somit steht die Forderung einer zweimal stetig differenzierbaren Referenz i. A. und insbesondere bei asymmetrischen Rändern im Konflikt zu unserer Definition der Mittellinie, sodass die Referenz zwangsläufig von der Mittellinie abweichen muss.

Nun sollen diejenigen Stützstellen  $M_k$  für eine Mittellinie berechnet werden, an denen die Symmetrie-Bedingung unter der Annahme abschnittsweise gerader Ränder erfüllt ist. Die Punkte des rechten Rands werden mit  $R$  bezeichnet, die des linken mit  $L$ . Der Index  $i$  zählt die rechten Randpunkte,  $j$  die linken und  $k$  die Punkte der Mittellinie. Falls möglich wird für jedes Segment  $\overline{R_i R_{i+1}}$  ein  $M_k$  berechnet, sodass  $k = i$  ist (ebenso gut könnte auch  $\overline{L_j L_{j+1}}$  verwendet werden). Die Idee hinter der segmentbasierten anstelle einer punktwisen Berechnung ist, die Berechnung der Mittellinie segmentweise auf die Berechnung der Winkelhalbierenden zurückzuführen. Es resultiert daraus eine Stützstelle  $M_k$ , die ungefähr mittig im Segment liegt, und in deren Umgebung die Mittellinie der Winkelhalbierenden entspricht. Hierfür wird zunächst ein Interpolationspunkt

$$\vec{p}_k = \frac{1}{2} (\vec{r}_i + \vec{r}_{i+1}) \quad (6.5)$$

gebildet, für den  $M_k$  so bestimmt werden soll, dass er auf der Winkelhalbierenden liegt, sodass

$$|\overline{M_k P_k}| \stackrel{!}{=} |\overline{M_k Q_k^*}| \quad (6.6)$$

erfüllt ist, wobei  $Q_k^*$  auf  $\overline{L_j L_{j+1}}$  liegt. Zunächst gehen wir davon aus, dass die Zuordnung

$$f_{P \rightarrow L} : k \mapsto j, \quad (6.7)$$

mit der jedem Punkt  $P_k$  ein passendes Segment  $\overline{L_j L_{j+1}}$  zugeordnet werden kann, bekannt sei. Nun wird der Vektor

$$\vec{d}_k := \frac{1}{\lambda} (\vec{q}_k^* - \vec{p}_k), \quad |\vec{d}_k| \stackrel{!}{=} 1, \quad (6.8)$$

als normierter Richtungsvektor von  $P_k$  zu dem gesuchten Punkt  $Q_k^*$  definiert. Er ist um  $90^\circ + \eta$  zum rechten Rand gedreht, d. h.

$$\frac{\vec{r}_{d,i} \cdot \vec{d}_k}{|\vec{r}_{d,i}|} = \cos(90^\circ + \eta) = -\sin(\eta) \quad \text{mit} \quad \vec{r}_{d,i} := \vec{r}_{i+1} - \vec{r}_i. \quad (6.9)$$

Der Winkel  $\eta$  ergibt sich darin aus

$$\cos(2\eta) = \frac{\vec{r}_{d,i} \cdot \vec{l}_{d,j}}{|\vec{r}_{d,i}| |\vec{l}_{d,j}|} \quad \text{mit} \quad \vec{l}_{d,j} := \vec{l}_{j+1} - \vec{l}_j. \quad (6.10)$$

Wenn die Fahrbahn wie in Bild 6.6 mit zunehmendem  $s$  breiter wird, ist per Definition  $\eta > 0$ . Das Vorzeichen von  $\eta$  ist also

$$\text{sgn}(\eta) = \text{sgn} \left( \vec{e}_3 \cdot (\vec{r}_{d,i} \times \vec{l}_{d,j}) \right). \quad (6.11)$$

Um den Richtungsvektor  $\vec{d}_k$  nun zu bestimmen, wird aus (6.9) und (6.8) das Gleichungssystem

$$\frac{r_{d,i,x} d_{k,x} + r_{d,i,y} d_{k,y}}{\sqrt{r_{d,i,x}^2 + r_{d,i,y}^2}} = -\sin(\eta) \quad \wedge \quad d_{k,x}^2 + d_{k,y}^2 = 1 \quad (6.12)$$

erstellt, welches mehrere Lösungen für  $\vec{d}_k = [d_{k,x} \ d_{k,y}]^T$  hat. Fordert man zusätzlich, dass  $\vec{d}_k$  in Richtung des linken Rands zeigen soll, also

$$\vec{e}_3 \cdot (\vec{r}_{d,i} \times \vec{d}_k) = r_{d,i,x} d_{k,y} + r_{d,i,y} d_{k,x} \geq 0 \quad (6.13)$$

ergibt sich die eindeutige Lösung

$$\vec{d}_k = \frac{1}{|\vec{r}_{d,i}|} \begin{bmatrix} -r_{d,i,x} \sin(|\eta|) - r_{d,i,y} \cos(\eta) \\ -r_{d,i,y} \sin(|\eta|) + r_{d,i,x} \cos(\eta) \end{bmatrix}. \quad (6.14)$$

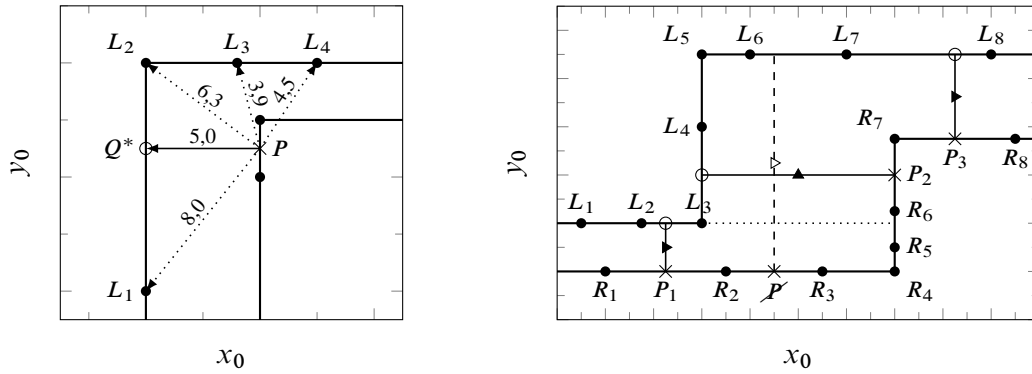
Für parallele Ränder ist  $\eta = 0$  und  $\vec{d}_k$  somit der Normalenvektor zu  $\vec{r}_{d,i}$ . Mit dem Richtungsvektor  $\vec{d}_k$  ergibt sich nun  $Q_k^*$  mit Hilfe von (6.8) aus dem Schnittpunkt der beiden Geraden

$$g_k : \vec{x} = \vec{p}_k + \lambda \vec{d}_k \quad \text{und} \quad h_k : \vec{x} = \vec{l}_j + \rho \vec{l}_{d,j}. \quad (6.15)$$

Der Schnittpunkt kann analog zu (6.4) berechnet werden. Sind alle Annahmen erfüllt, müssen  $\lambda > 0$  und  $\rho \in [0, 1]$  sein. Die Lösung für  $\rho$  wird zur Kontrolle verwendet: Wenn  $\rho \notin [0, 1]$  ist, ist das Segment  $\overline{L_j L_{j+1}}$  ungeeignet, d. h. die Zuordnung (6.7) ist nicht korrekt. Bei  $\rho < 0$  muss i. d. R. ein zurückliegender Abschnitt verwendet werden, für  $\rho > 1$  ein weiter vorne liegender. Für das Segment  $\overline{R_{i-1} R_i}$  aus Bild 6.6 kann mit dieser Methode kein Mittelpunkt bestimmt werden. Denn wenn man es einsetzt, ergibt sich für das Segment  $\overline{L_{j-1} L_j}$  ein Wert  $\rho > 1$  und für  $\overline{L_j L_{j+1}}$  ein Wert  $\rho < 0$ . Falls aber ein passender Abschnitt vorliegt, lautet der gesuchte Mittelpunkt

$$\vec{m}_k = \vec{p}_k + \frac{1}{2} \lambda_k \vec{d}_k. \quad (6.16)$$

Um eine möglichst gute Zuordnung (6.7) für einen Punkt  $P_k$  zu erhalten, wird eine Auswahl an Kandidaten bestimmt, nämlich alle Segmente  $\overline{L_j L_{j+1}}$ , deren Orientierung relativ zum Segment  $\overline{R_i R_{i+1}}$  weniger als ca.  $70^\circ - 80^\circ$  beträgt. Anschließend wird zuerst der Kandidat verwendet, der den geringsten Euklidischen Abstand zu  $P_k$  hat. Würde keine Vorauswahl über die Orientierung getroffen, sondern nur der Abstand verwendet, könnten leicht Fehlzuordnungen auftreten, wie Bild 6.7(a) zeigt. Sie würden allerdings beim Lösen des Gleichungssystems (6.15) auffallen. Eine andere Fehlzuordnung, die selbst dann nicht auffällt, zeigt Bild 6.7(b): Zum Punkt  $P_1$  ist der Mittelpunkt gut zu bestimmen. Für das folgende Segment wird  $\nearrow$  entlang der gestrichelten Linie  $\overline{L_6 L_7}$  zugeordnet. Das Problem ist, dass die Orientierung der zugeordneten Segmente zwar perfekt passt, die Zuordnung aber dennoch zu einem völlig falschen und verschlungenen Verlauf der Mittellinie führt, wenn man die Positionen der nachfolgenden Punkte berücksichtigt (zur besseren Übersicht sind nicht alle eingezeichnet). Ein Indiz für ein derartiges Problem ist die Verwendung der linken Randpunkte in nicht monotoner Reihenfolge. Geometrisch betrachtet entsteht das Problem dadurch, dass der linke Rand an  $L_3$  im rechten (oder spitzen) Winkel nach links abknickt.



(a) Das gegenüberliegende Segment zu  $P$  kann nicht über den Abstand, sondern nur die Orientierung ermittelt werden.

(b) Wenn der linke Rand durch eine orthogonale oder spitze Ecke konkav ist, werden Punkte im rechten Rand übersprungen, um Fehlzuordnungen zu vermeiden.

**Bild 6.7:** Bei der Zuordnung gegenüberliegender Randsegmente sind Spezialfälle zu beachten.

Wenn eine solche Ecke zwischen dem zuletzt zugeordneten Segment (also  $\overline{L_2 L_3}$ ) und dem aktuell zugeordneten (also  $\overline{L_6 L_7}$ ) liegt, muss dazwischen ein Abschnitt liegen, dessen Orientierung nicht zum rechten Randabschnitt passt. Der Algorithmus projiziert dann den Eckpunkt  $L_3$  auf den rechten Rand und überspringt alle Punkte in diesem Bereich, also hier von  $R_2$  bis  $R_6$ . Die dadurch entstehende Lücke kann gut mit einem HYMAN-Spline [42] interpoliert werden.

#### 6.1.4 Krümmungsabhängige Abtastung der Referenzlinie

Bevor der folgende Abschnitt beschreibt, wie die Referenzlinie durch Filterung der Mittellinie entsteht, geht dieser Abschnitt auf die Diskretisierung  $h_k = \Delta s_k, k = 1, \dots, N$ , der Referenzlinie ein. Sie erfolgt während der Filterung auf Basis eines heuristischen Ansatzes. Um das Optimierungsproblem klein zu halten ohne dabei nennenswerte Einbußen im Detaillierungsgrad der Lösung zu haben, gilt es, die Diskretisierung so grob wie möglich und so fein wie nötig zu wählen. Da sie im Ortsbereich und ohne Kenntnis der Trajektorie erfolgen muss, können hierfür keine dynamischen Modelleigenschaften, wie Zeitkonstanten herangezogen werden. Es wird angenommen, dass die Dynamik in der Nähe von Kurven höher als auf Geraden ist und zunimmt, je enger die Kurve ist. Diese Annahme führt zu einer antiproportionalen Abhängigkeit zwischen Krümmung  $\kappa$  und gewünschter Schrittweite  $h^*$ , welche durch

$$(h^* \circ f)(\kappa) = \frac{h_{\max}}{\left(\frac{h_{\max}}{h_c} - 1\right) \left(\frac{f(\kappa)}{f_c}\right)^{1/\zeta} + 1} \quad (6.17)$$

mit  $f(\kappa)$  : zusätzliche Hilfsfunktion, im einfachsten Fall  $f(\kappa) = |\kappa|$

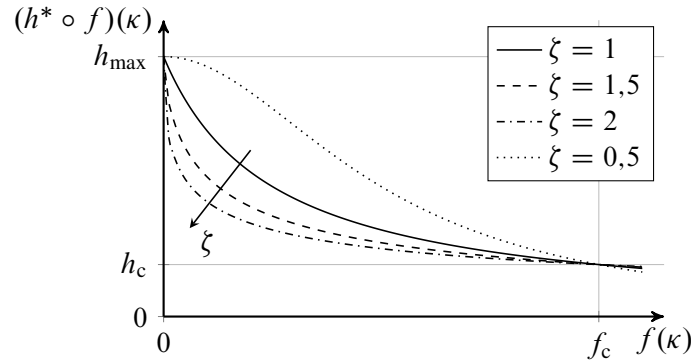
$h_{\max}$  : maximale Schrittweite in m

$f_c$  : Funktionswert, bei dem  $h^*$  auf  $h_c < h_{\max}$  gesunken ist ( $f_c > 0$ )

$h_c$  : Schrittweite bei  $f(\kappa) = f_c$

$\zeta$  : inverser Exponent zur Beeinflussung des Verlaufs, siehe Bild 6.8





**Bild 6.8:** Abhängigkeit des Verlaufs der Diskretisierungskurve vom Parameter  $\zeta$

heuristisch beschrieben wird und zu einem Verlauf nach Bild 6.8 führt. In der Funktion wird also die maximale Schrittweite  $h_{\max}$  für  $\kappa = 0$  und eine weitere Schrittweite  $h_c$  für  $f(\kappa) = f_c$  definiert. Meist wird für  $h_c$  die minimale Schrittweite und für  $f_c = f(\max(|\kappa|))$  die maximal vorkommende Krümmung angesetzt. Über den Parameter  $\zeta$  wird appliziert, wie stark die Schrittweite mit zunehmender Krümmung abnimmt, siehe Bild 6.8.

Oft enthält die Trajektorie auch vor und hinter den Kurven Bereiche, die eine feinere Diskretisierung erfordern. Mit einer Hilfsfunktion  $f(\kappa) = |\kappa|$  sind diese Bereiche zu grob diskretisiert. Daher werden für  $f$  zusätzlich die ersten Ableitungen von  $\kappa$  berechnet. Anschließend werden  $\kappa, \kappa'$  etc. normiert und mit einem Tiefpass gefiltert, damit sich Peaks in den Signalen auf die umliegenden Bereiche auswirken. Für die Funktion  $f$  wird dann von allen (gefilterten und ungefilterten) Signalen das Betragsmaximum verwendet. So bleiben Maxima unverändert erhalten, jedoch werfen sie quasi einen zusätzlichen Schatten auf ihre Umgebung.

Aus der Zielschrittweite  $h^*(s)$  werden die finalen Stützstellen  $s_k, k = 1, 2, \dots$  iterativ ermittelt:

$$s_1 = 0 \quad (6.18a)$$

$$s_{k+1} = s_k + \min_{s_k \leq \tilde{s} < s_k + h^*(s_k)} h^*(\tilde{s}), \quad k \geq 1, \quad s_{k+1} \leq S_{\text{end}} \quad (6.18b)$$

Am Streckenende wird für eine gewisse Anzahl an Stützstellen die Schrittweite  $h^*$  so reduziert, dass gerade  $s_{N+1} = S_{\text{end}}$  ist, ohne dabei eine minimal vorgegebene Schrittweite zu unterschreiten.

## 6.2 Ein Kappa-Filter zur Glättung parametrierter Kurven

Das im Folgenden beschriebene „Kappa-Filter“ dient dazu, eine mindestens einmal stetig differenzierbare Kurve  $\vec{P}(s) \in \mathbb{R}^2$  zu glätten, welche über ihre Bogenlänge  $s$  parametrisiert ist:

$$\vec{P}(s) = [\mathcal{P}_x(s) \quad \mathcal{P}_y(s)]^\top \quad (6.19)$$

Die Kurve sei hier durch einen Spline beschrieben, welcher durch die Stützstellen  $P_1, \dots, P_N$  verlaufe, sodass  $\vec{p}_k = \vec{P}(s_k), k = 1, \dots, N$ , der Ortsvektor zu  $P_k$  sei. Für die eigentliche Op-

timierung, die im folgenden Abschnitt beschrieben wird, sind allerdings nur die Stützstellen relevant, und der Spline wird lediglich zur Bestimmung der Bogenlänge sowie der Orientierung an den Stützstellen (Tangenten- bzw. Normalenvektoren) verwendet.

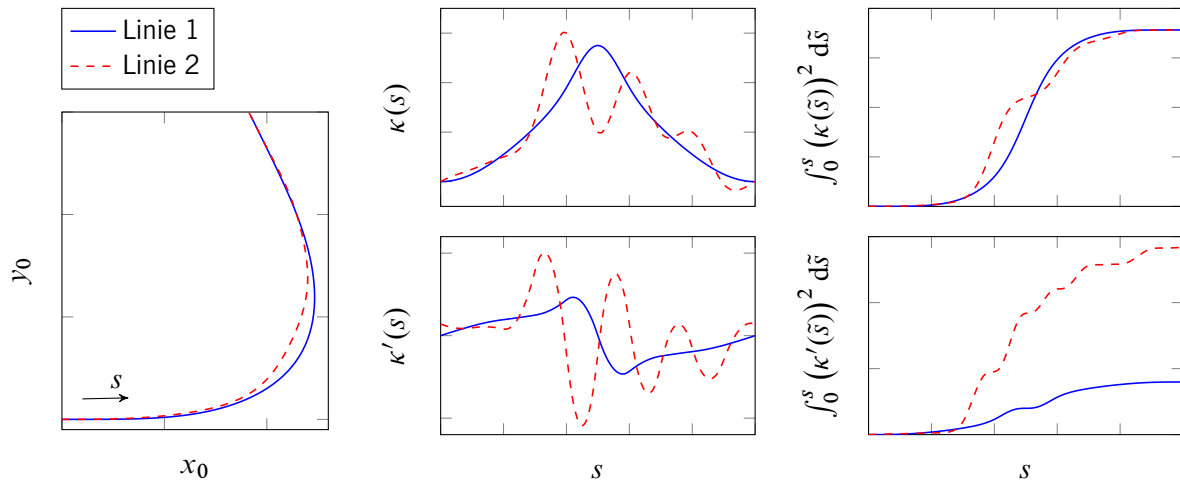
Das Kappa-Filter soll im Ergebnis zu einem Krümmungsverlauf der Kurve führen, der ähnlich aussieht, als hätte man die Krümmung phasenfrei tiefpassgefiltert. Dabei darf sich die Kurve selbst an jeder Stützstelle nur um einen vorgegebenen Maximalwert verschieben. Wie wir im folgenden Abschnitt sehen werden, lässt sich das Kappa-Filter parametrieren, sodass die Minimierung des Krümmungsbetrags und die Minimierung der Krümmungsänderung gegeneinander gewichtet werden. Diese Kombination bewirkt eine sehr gute Glättung der Linie. Im Zusammenhang der vorliegenden Arbeit gibt es hierfür zwei Anwendungen:

1. Damit die Referenzlinie eine stetige Krümmung aufweist, muss sie mindestens zweimal stetig differenzierbar sein. In Abschnitt 6.1.3 hat sich gezeigt, dass eine Mittellinie diese Bedingung auch bei stetigen Rändern i. A. nicht erfüllen kann, und es daher sinnvoll ist, die Linie zu glätten. Zudem fließt die Krümmung über  $\sigma$  in die Modellgleichung (3.35) auf S. 36 ein und wird aufgrund der Diskretisierung (5.4) auf S. 66 während eines Abtastschritts als konstant angenommen. Um diese Annahme treffen zu können, sollte die Krümmungsänderung gering gehalten werden. Bedenkt man zudem, dass der Streckenverlauf ausschließlich über die Krümmung in das Optimierungsproblem eingeht, ist damit zu rechnen, dass eine lokal ungenaue Krümmung zu einer lokal suboptimalen Fahrlinie führt, was sich aber erst nach ihrer Rücktransformation (meist im realen Fahrversuch) zeigt. Je glatter die Referenzlinie ist, umso genauer lässt sich ihre Krümmung bestimmen. Für diese Anwendung wurde das Kappa-Filter in einem früheren Entwicklungsstadium bereits in [96] beschrieben.
2. In Abschnitt 2.2 wurde beschrieben, dass krümmungsminimale Linien oftmals eine gute Approximation für zeitoptimale Linien darstellen, wie es auch in [12, 37] genutzt wird. In dem Beispiel in Bild 2.3 auf S. 16 wurde die krümmungsminimale Linie für den Vergleich zur zeitoptimalen mit diesem Kappa-Filter berechnet.

Eine klassische Tiefpass-Filterung der Krümmung wäre nicht zielführend, weil sie nicht garantieren kann, dass die Verschiebung der Stützstellen innerhalb eines vorgegebenen Bereichs liegt. Eine Tiefpass-Filterung der kartesischen Koordinaten ist ebenso wenig zielführend, weil es kaum möglich ist, eine einheitliche Grenzfrequenz zu bestimmen, die zum einen auch niederfrequente Schwingungen reduziert, um den Verlauf ausreichend zu glätten, und zum anderen aber die Kurven nicht zu stark schneidet. Daher wird die Filterung als Optimierungsproblem formuliert.

### 6.2.1 Filterung durch Optimierung

Das Ziel dieser Optimierung ist es, das Integral der Krümmung  $\kappa(s)$  der Linie  $\vec{\mathcal{P}}(s)$  sowie der 1. und 2. Ableitung von  $\kappa$  zu minimieren und dabei die Verschiebung  $e(s)$  der Linie zu begrenzen. Es ist wichtig, die Ableitungen mit zu berücksichtigen, wie Bild 6.9 am Beispiel zweier Linien für eine Linkskurve zeigt: Ihr Verlauf in Bild 6.9(a) ist ähnlich, doch unterscheidet sich ihre



(a) Der Linienvorlauf in kartesischen Koordinaten ist bei beiden Linien ähnlich. (b) Linie 2 ist viel unruhiger als Linie 1. Dennoch ist das Integral der quadrierten Krümmung gleich groß. Erst das Integral der quadrierten Krümmungsableitung zeigt, wie unruhig Linie 2 ist.

**Bild 6.9:** Zwei Linien mit ähnlichem Verlauf aber sehr unterschiedlicher Krümmung können zum selben Wert im Integral der quadrierten Krümmung führen.

Krümmung in Bild 6.9(b) (oben links) stark. Obwohl Position und Ausrichtung am Anfang und Ende der Strecke vorgegeben sind, ist Linie 1 viel glatter als Linie 2, welche die Kurve in mehrere Teilkurven aufteilt. Dennoch ist das Integral von  $\kappa^2$  am Streckenende beinahe identisch.

Würde man nur anhand dieses Integrals eine *krümmungsminimale Linie* entwerfen, wäre Linie 2 also gleich gut wie Linie 1. Betrachtet man hingegen auch das Integral der quadrierten Ableitung von  $\kappa$  (Bild 6.9(b) unten), ist Linie 1 deutlich besser als Linie 2. Für einen Verlauf, der unserer Vorstellung einer glatten Linie entspricht, müssen also beide Integrale möglichst klein sein.

Für eine über  $s$  parametrisierte Linie  $\vec{\mathcal{P}}(s)$  ist die Krümmung (vgl. [53, S. 503])

$$\kappa(s) = \frac{\mathcal{P}'_x(s) \mathcal{P}''_y(s) - \mathcal{P}''_x(s) \mathcal{P}'_y(s)}{|\vec{\mathcal{P}}'(s)|^3}. \quad (6.20)$$

Parametrisiert man sie über ihre eigene Bogenlänge, also  $ds = \sqrt{dx^2 + dy^2}$ , steht im Nenner

$$|\vec{\mathcal{P}}'| = \frac{\sqrt{(d\mathcal{P}_x)^2 + (d\mathcal{P}_y)^2}}{ds} = \frac{\sqrt{dx^2 + dy^2}}{\sqrt{dx^2 + dy^2}} = 1, \quad (6.21)$$

was analog zu (3.13) auf S. 29 ist. Unter dieser Voraussetzung ergeben sich einfachere Terme für die Krümmung und ihre Ableitungen, wobei im Folgenden zur besseren Übersicht  $x$  statt  $\mathcal{P}_x$  und  $y$  statt  $\mathcal{P}_y$  geschrieben wird:

$$\kappa = x^{(1)}y^{(2)} - x^{(2)}y^{(1)} \quad (6.22a)$$

$$\kappa^{(1)} = x^{(2)}y^{(2)} + x^{(1)}y^{(3)} - x^{(3)}y^{(1)} - x^{(2)}y^{(2)} = x^{(1)}y^{(3)} - x^{(3)}y^{(1)} \quad (6.22b)$$

$$\kappa^{(2)} = x^{(2)}y^{(3)} + x^{(1)}y^{(4)} - x^{(4)}y^{(1)} - x^{(3)}y^{(2)} \quad (6.22c)$$

Im Optimum sollen alle 3 Terme in (6.22) betragsmäßig klein sein. Würde man die gewichtete Summe ihrer Quadrate direkt minimieren, entstünde ein komplexes hochgradig nichtlineares Optimierungsproblem. Für eine höhere Effizienz wird ein quadratisches Ersatzproblem formuliert, welches das Optimierungsziel annähert. Um die Terme in (6.22) umzuformen, nutzen wir ihre Parametrierung über der Bogenlänge, nämlich

$$(x')^2 + (y')^2 = 1 \quad \Rightarrow \quad \begin{cases} (x')^2 = 1 - (y')^2, \\ (y')^2 = 1 - (x')^2. \end{cases} \quad (6.23)$$

Differentiation von (6.23) liefert zudem

$$2x'x'' + 2y'y'' = 0 \quad \Leftrightarrow \quad x'x'' = -y'y''. \quad (6.24)$$

Das Quadrat von (6.22a) lässt sich äquivalent umformen in

$$\begin{aligned} \kappa^2 &= (x'y'' - x''y')^2 \\ &= (x')^2 (y'')^2 - 2x'x''y'y'' + (x'')^2 (y')^2 \\ &\stackrel{(6.23)}{=} (y'')^2 - (y')^2 (y'')^2 - 2x'x''y'y'' + (x'')^2 - (x')^2 (x'')^2 \\ &= (x'')^2 + (y'')^2 - (x'x'' + y'y'')^2 \\ &\stackrel{(6.24)}{=} (x'')^2 + (y'')^2, \end{aligned} \quad (6.25)$$

was später zu einem rein quadratischen Gütemaß führt. Analog lässt sich zeigen, dass (6.22b) zu

$$(\kappa')^2 = (x''')^2 + (y''')^2 - (x'x''' + y'y''')^2 \quad (6.26)$$

führt. Da alle quadratischen Terme positiv sind, muss  $(x''')^2 + (y''')^2 \geq (x'x''' + y'y''')^2$  sein, sodass sich  $(\kappa')^2$  durch  $(x''')^2 + (y''')^2$  nach oben abschätzen lässt. Wenn die ersten beiden Summanden in (6.26) minimal sind, wird zudem auch der dritte Summand betragsmäßig klein sein, sodass er im Ersatzproblem vernachlässigt wird. Für die 2. Ableitung  $\kappa''$  in (6.22c) ergeben sich die 4. Ableitungen von  $x$  und  $y$  zuzüglich mehrerer Terme mit gemischten Ableitungen, deren Abschätzung schwierig ist. Für das Ersatzproblem werden sie dennoch vernachlässigt, weil  $\kappa''$  in den meisten Fällen durch  $x^{(4)}$  und  $y^{(4)}$  dominiert wird, wenn – wie im vorliegenden Fall annehmen – sowohl  $\kappa$  als auch  $\kappa'$  klein ist.

Die Parametrierung über der Bogenlänge erlaubt also eine immense Vereinfachung. So kann im Gegensatz zu [37, S. 12] auch der Term  $2x'x''y'y''$  in (6.25) entfallen, was mittels Äquivalenzumformungen gezeigt wurde.

Bevor nun anhand des Optimierungsziels ein konkretes Optimierungsproblem formuliert werden kann, muss zunächst geklärt werden, auf welche Art und Weise eine gegebene Linie  $\vec{P}$  durch die Optimierung modifiziert werden soll. Da das Problem später numerisch gelöst werden soll, betrachten wir  $\vec{P}$  nur an seinen  $N$  Stützstellen  $\vec{p}_1, \dots, \vec{p}_N$ . Um zum einen den Abstand zwischen zwei Stützstellen möglichst wenig zu verändern und zum anderen ein Maß für die Abweichung zwischen der Ausgangslinie und der finalen Linie zu haben, darf jede Stützstelle nur entlang ihres



Relevanz, es bietet sich aber an, sie in Hinblick auf andere Anwendungen gleich mit ins Optimierungsproblem aufzunehmen, da es nur geringfügigen Mehraufwand mit sich bringt.

Im nächsten Schritt müssen die Ableitungen  $x^{(i)}$  und  $y^{(i)}$  bestimmt werden. Wenn  $\vec{\mathcal{P}}$  hinreichend oft analytisch differenzierbar ist, können die analytisch abgeleiteten Terme ins Optimierungsproblem aufgenommen werden. Gehen wir aber davon aus, dass  $\vec{\mathcal{P}}$  durch einen kubischen Spline dargestellt ist, ist bereits die 3. Ableitung konstant und somit für eine Optimierung ungeeignet. Zudem ist sie selbst im Fall eines natürlichen Splines nicht mehr stetig. In [37, 12] werden die Koeffizienten des kubischen natürlichen Splines verwendet, weil dort die Ableitungen der Krümmung nicht eingehen. Hier werden die Ableitungen  $n$ -ter Ordnung jedoch über Differenzenquotienten  $x^{(n)}(s_k) \approx \frac{\Delta^n x_k}{\Delta s_k^n}$  approximiert, welche genauso wie in (5.13) ff. auf S. 71 mittels der Matrix  $\mathbf{D}_n$  für alle Stützstellen  $\mathbf{x} = [x_1 \ \dots \ x_N]^\top$  (und ebenso für  $\mathbf{y}$ ) berechnet werden:

$$\mathbf{x}^{(n)} := \begin{bmatrix} \frac{\Delta^n x_1}{\Delta s_1^n} \\ \vdots \\ \frac{\Delta^n x_N}{\Delta s_N^n} \end{bmatrix} = \mathbf{D}_n \mathbf{x} \quad \Rightarrow \quad \sum_{k=1}^N \left(x_k^{(n)}\right)^2 = \mathbf{x}^\top \mathbf{D}_n^\top \mathbf{D}_n \mathbf{x} \quad (6.31)$$

Um Oszillationen in der Lösung zu vermeiden, empfiehlt sich auch hier wiederum die Verwendung eines reinen Vorwärts- oder Rückwärts-Differenzenquotienten anstelle eines zentralen Differenzenquotienten, was in Anhang C begründet ist. Ob man für  $\mathbf{D}$  nun einen Vorwärts- oder einen Rückwärts-Differenzenquotienten verwendet, ist für diese Anwendung irrelevant, weil das Ergebnis in  $\mathbf{Q} := \mathbf{D}^\top \mathbf{D}$  dasselbe ist.

Die Elemente der Matrix  $\mathbf{D}$  hängen von den Distanzen

$$\Delta s_k = \int_{s_k}^{s_{k+1}} \sqrt{\left(\frac{dx(\tilde{s})}{d\tilde{s}}\right)^2 + \left(\frac{dy(\tilde{s})}{d\tilde{s}}\right)^2} d\tilde{s} \approx \sqrt{(\Delta x_k)^2 + (\Delta y_k)^2} \quad (6.32)$$

zwischen den Stützstellen ab. Dass sie nicht entlang der Linie für alle  $k$  gleich sind, stellt kein Problem dar, jedoch verändern sie sich während der Optimierung durch die Verschiebung der Stützstellen. Von einer analytischen Berücksichtigung von (6.32) im Optimierungsproblem ist abzuraten, weil es dadurch stark nichtlinear und sehr komplex wird. Denkbar wäre es,  $\Delta s_k$  im Optimierungsproblem analytisch durch eine affine Funktion von  $\mathbf{e}$  zu approximieren. Weil  $\Delta s_k$  aber bei den Elementen von  $\mathbf{D}$  im Nenner und in unterschiedlichen Potenzen auftritt, entsteht auch dabei ein nichtlineares Gütemaß. Aufgrund dieser Problematik wird für den Spline der Ausgangslinie

$$\Delta s_k = \int_{s_k}^{s_{k+1}} \sqrt{\left(\frac{d\mathcal{P}_x(\tilde{s})}{d\tilde{s}}\right)^2 + \left(\frac{d\mathcal{P}_y(\tilde{s})}{d\tilde{s}}\right)^2} d\tilde{s} = \text{const}_k \quad (6.33)$$

berechnet und während der Optimierung als fix angenommen. So behält das Problem die Struktur eines QPs, das sich effizient lösen lässt. Nach Ablauf der Optimierung wird  $\Delta s_k$  neu berechnet und ggf. eine weitere Optimierung gestartet. Die Praxis zeigt, dass sich die Linie oft schon nach

2 Iterationen beim Glätten und 4 Iterationen für Rennlinienberechnungen kaum noch verändert. Eine solche iterative Strategie kommt auch in [37, S. 16] zum Einsatz.

Schließlich kann die quadratische Gütefunktion mit den Gewichtungsfaktoren  $\lambda_{\dots} \geq 0$

$$\begin{aligned}\Phi_{\text{SKF}} &= \frac{1}{2} \sum_{k=1}^N \left( \sum_{i=1}^4 \lambda_i^2 \left( \left( \frac{\Delta^i x_k}{\Delta s_k^i} \right)^2 + \left( \frac{\Delta^i y_k}{\Delta s_k^i} \right)^2 \right) + \lambda_e^2 e_k^2 \right) \\ &= \frac{1}{2} [\mathbf{e}^\top \quad \mathbf{x}^\top \quad \mathbf{y}^\top] \begin{bmatrix} \mathbf{Q}_e & & \\ & \mathbf{Q}_{xy} & \\ & & \mathbf{Q}_{xy} \end{bmatrix} \begin{bmatrix} \mathbf{e} \\ \mathbf{x} \\ \mathbf{y} \end{bmatrix}\end{aligned}\quad (6.34)$$

aufgestellt werden, worin  $\mathbf{Q}_e = \lambda_e^2 \mathbf{I}_N$  und  $\mathbf{Q}_{xy} = \sum_{i=1}^4 \lambda_i^2 \mathbf{D}_i^\top \mathbf{D}_i$  ist.

Es bietet sich nun an, die Gleichungsnebenbedingung (GNB) aus (6.30) in die Gütefunktion (6.34) einzusetzen, um  $\mathbf{x}$  und  $\mathbf{y}$  als Optimierungsvariablen zu eliminieren, da die Linie durch  $\mathbf{e}$  und  $[\mathbf{p}_x \quad \mathbf{p}_y] = \text{const}$  bereits eindeutig beschrieben ist. Das Einsetzverfahren wurde in Abschnitt 4.3.1 auf S. 56 als Methode zum Behandeln von GNB beschrieben. Hierfür definieren wir zunächst das Teil-Gütemaß  $\Phi_{\text{SKF},\kappa}$  über

$$\Phi_{\text{SKF}} = \frac{1}{2} \begin{bmatrix} \mathbf{e} \\ \mathbf{x} \\ \mathbf{y} \end{bmatrix}^\top \begin{bmatrix} \mathbf{Q}_e & & \\ & \mathbf{Q}_{xy} & \\ & & \mathbf{Q}_{xy} \end{bmatrix} \begin{bmatrix} \mathbf{e} \\ \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \frac{1}{2} \mathbf{e}^\top \mathbf{Q}_e \mathbf{e} + \underbrace{\frac{1}{2} [\mathbf{x}^\top \quad \mathbf{y}^\top] \begin{bmatrix} \mathbf{Q}_{xy} & \\ & \mathbf{Q}_{xy} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}}_{=: \Phi_{\text{SKF},\kappa}}, \quad (6.35)$$

worin sich

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{p}_x \\ \mathbf{p}_y \end{bmatrix} + \begin{bmatrix} \mathbf{R}_x \\ \mathbf{R}_y \end{bmatrix} \mathbf{e} \quad \text{mit} \quad \begin{bmatrix} \mathbf{R}_x \\ \mathbf{R}_y \end{bmatrix} := \begin{bmatrix} \text{diag}(\mathbf{n}_x) \\ \text{diag}(\mathbf{n}_y) \end{bmatrix} \in \mathbb{R}^{2N \times N} \quad (6.36)$$

gemäß (6.28) angeben lässt. Für die folgende Berechnung nutzen wir die Symmetrie der quadratischen Matrizen  $\mathbf{Q}_{xy}$ ,  $\mathbf{R}_x$  und  $\mathbf{R}_y$ , also dass  $\mathbf{Q}_{xy}^\top = \mathbf{Q}_{xy}$  ist etc. Es folgt:

$$\Phi_{\text{SKF},\kappa} = \left( \begin{bmatrix} \mathbf{p}_x^\top & \mathbf{p}_y^\top \end{bmatrix} + \mathbf{e}^\top \begin{bmatrix} \mathbf{R}_x & \mathbf{R}_y \end{bmatrix} \right) \begin{bmatrix} \mathbf{Q}_{xy} & \\ & \mathbf{Q}_{xy} \end{bmatrix} \left( \begin{bmatrix} \mathbf{p}_x \\ \mathbf{p}_y \end{bmatrix} + \begin{bmatrix} \mathbf{R}_x \\ \mathbf{R}_y \end{bmatrix} \mathbf{e} \right) \quad (6.37a)$$

$$\begin{aligned}&= \mathbf{e}^\top \begin{bmatrix} \mathbf{R}_x & \mathbf{R}_y \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{xy} & \\ & \mathbf{Q}_{xy} \end{bmatrix} \begin{bmatrix} \mathbf{R}_x \\ \mathbf{R}_y \end{bmatrix} \mathbf{e} + \dots \\ &+ \mathbf{e}^\top \begin{bmatrix} \mathbf{R}_x & \mathbf{R}_y \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{xy} & \\ & \mathbf{Q}_{xy} \end{bmatrix} \begin{bmatrix} \mathbf{p}_x \\ \mathbf{p}_y \end{bmatrix} + \begin{bmatrix} \mathbf{p}_x^\top & \mathbf{p}_y^\top \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{xy} & \\ & \mathbf{Q}_{xy} \end{bmatrix} \begin{bmatrix} \mathbf{R}_x \\ \mathbf{R}_y \end{bmatrix} \mathbf{e} + \dots\end{aligned}\quad (6.37b)$$

$$\begin{aligned}&+ \underbrace{\begin{bmatrix} \mathbf{p}_x^\top & \mathbf{p}_y^\top \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{xy} & \\ & \mathbf{Q}_{xy} \end{bmatrix} \begin{bmatrix} \mathbf{p}_x \\ \mathbf{p}_y \end{bmatrix}}_{\text{const}} \\ &= \underbrace{\mathbf{e}^\top \begin{bmatrix} \mathbf{R}_x & \mathbf{R}_y \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{xy} & \\ & \mathbf{Q}_{xy} \end{bmatrix} \begin{bmatrix} \mathbf{R}_x \\ \mathbf{R}_y \end{bmatrix} \mathbf{e}}_{\tilde{\mathbf{Q}}_{xy}} + 2 \underbrace{\begin{bmatrix} \mathbf{p}_x^\top & \mathbf{p}_y^\top \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{xy} & \\ & \mathbf{Q}_{xy} \end{bmatrix} \begin{bmatrix} \mathbf{R}_x \\ \mathbf{R}_y \end{bmatrix} \mathbf{e}}_{\mathbf{f}^\top} + \text{const}\end{aligned}\quad (6.37c)$$

Somit ergibt sich insgesamt die Gütefunktion

$$\Phi_{\text{SKF}} = \frac{1}{2} \mathbf{e}^\top (\mathbf{Q}_e + \tilde{\mathbf{Q}}_{xy}) \mathbf{e} + \mathbf{f}^\top \mathbf{e} + \text{const}, \quad (6.38)$$

wobei der konstante Anteil für die Optimierung irrelevant ist. Über eine einfache Box-Restriktion  $e_{k,\text{re}} \leq e_k \leq e_{k,\text{li}}$  besteht die Möglichkeit, die Verschiebung  $e$  für jeden Punkt so zu begrenzen, dass die resultierende Linie in einem definierten Schlauch liegt.

In [37] ist zusätzlich eine Nebenbedingung für die maximale Krümmung vorgesehen, damit die resultierende Linie fahrbar ist. Eine solche NB existiert hier nicht, weil die Optimierung ja per se die minimale Krümmung ausgibt, die ohne Verlassen der Strecke möglich ist. Und aufgrund der Krümmungsableitungen in der Gütefunktion treten bei sinnvoller Parametrierung auch keine problematischen Peaks in der Krümmung auf, wie in Bild 6.9 exemplarisch gezeigt wurde.

### 6.2.3 Ergebnisse des Kappa-Filters zum Glätten

Für den Streckenverlauf des *Autodroms Most* (vgl. Bild 5.9 auf S. 90) wird gemäß Abschnitt 6.1.3 eine Mittellinie berechnet, die anschließend mithilfe des Kappa-Filters geglättet wird. Für dieses Beispiel beträgt die zulässige Verschiebung der Linie gerade einmal  $\pm 5$  cm. Bild 6.10(a) zeigt für einen gewissen Streckenabschnitt den Krümmungsverlauf und seine (numerisch berechneten) Ableitungen sowohl der Ausgangslinie (links) als auch der optimierten Linie (rechts). Darunter in Bild 6.10(b) ist die Verschiebung  $e$  abgebildet, deren betragsmäßiger Mittelwert bei 2,3 cm liegt.<sup>2</sup>

Für die Optimierung wurde 1 Durchlauf mit 14 Iterationen durchgeführt. (Die Krümmung wurde danach natürlich für das tatsächliche  $\Delta s_k$  ermittelt.) Die Filterung der gesamten 4,2 km langen Strecke mit 2820 Punkten hat in MATLAB auf einer INTEL Core i7 Mobile CPU insgesamt knapp 70 ms gedauert, wovon die Optimierung selbst 47 ms benötigt hat. Die Implementierung des Kappa-Filters in MATLAB ist nicht so stark auf Rechenzeit getrimmt wie die Trajektorienplanung. Daher ließen sich mit konsequenter Codeoptimierung noch Verbesserungen erzielen.

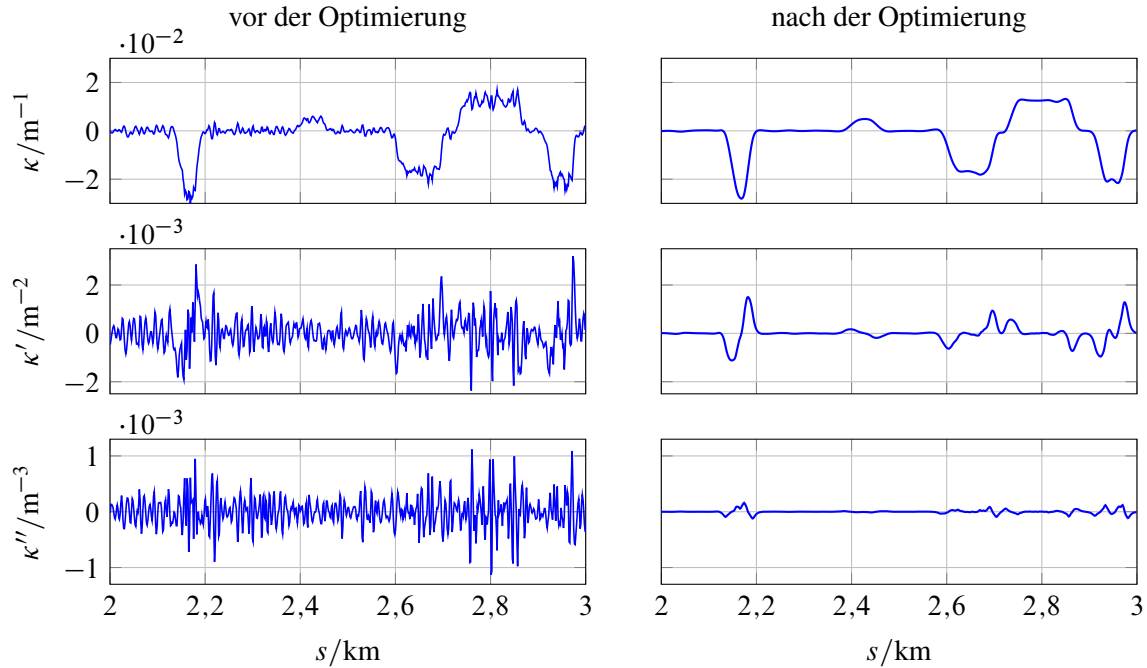
Zusammenfassend liefert das Kappa-Filter also eine hervorragend glatte Linie, die dennoch sehr nahe an der Ursprungslinie verläuft und den vorgegebenen Bereich nicht verlässt.

### 6.2.4 Ergebnisse des Kappa-Filters zur Rennlinien-Approximation

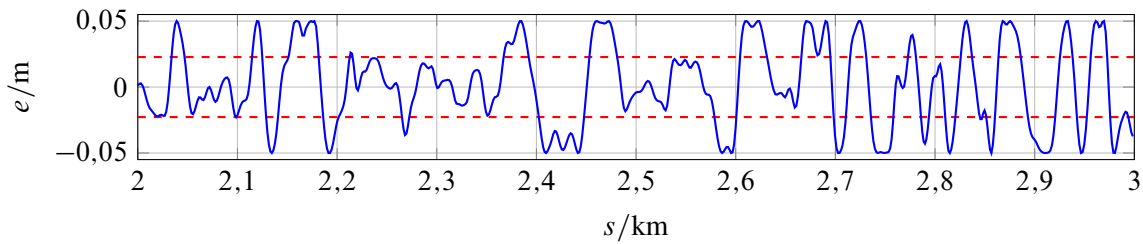
Ein weiteres Anwendungsgebiet dieses Kappa-Filters ist die Approximation von Rennlinien, da eine geringe Krümmung nach (2.1) bei gleicher Geschwindigkeit zu einer geringeren Zentripetalbeschleunigung führt, oder auch bei gleicher Zentripetalbeschleunigung die Geschwindigkeit erhöht werden kann. Dieser Abschnitt beurteilt die *krümmungsminimale* Linie anhand eines Vergleichs mit einer zeitoptimalen. In Bild 2.3 auf S. 16 konnte man bereits sehen, dass

<sup>2</sup>Eine Glättung desselben Streckenabschnitts wurde auch schon 2017 in [96, S. 231] veröffentlicht. Dort erfolgte die Optimierung allerdings noch mit einer äquidistanten Abtastung und die Verschiebung wurde nur auf  $\pm 15$  cm begrenzt, wodurch sich der Krümmungsverlauf im Detail unterscheidet.





(a) Der stark verrauschte Krümmungsverlauf der Ausgangslinie ist nach der Optimierung bis in die zweite Ableitung sehr glatt.

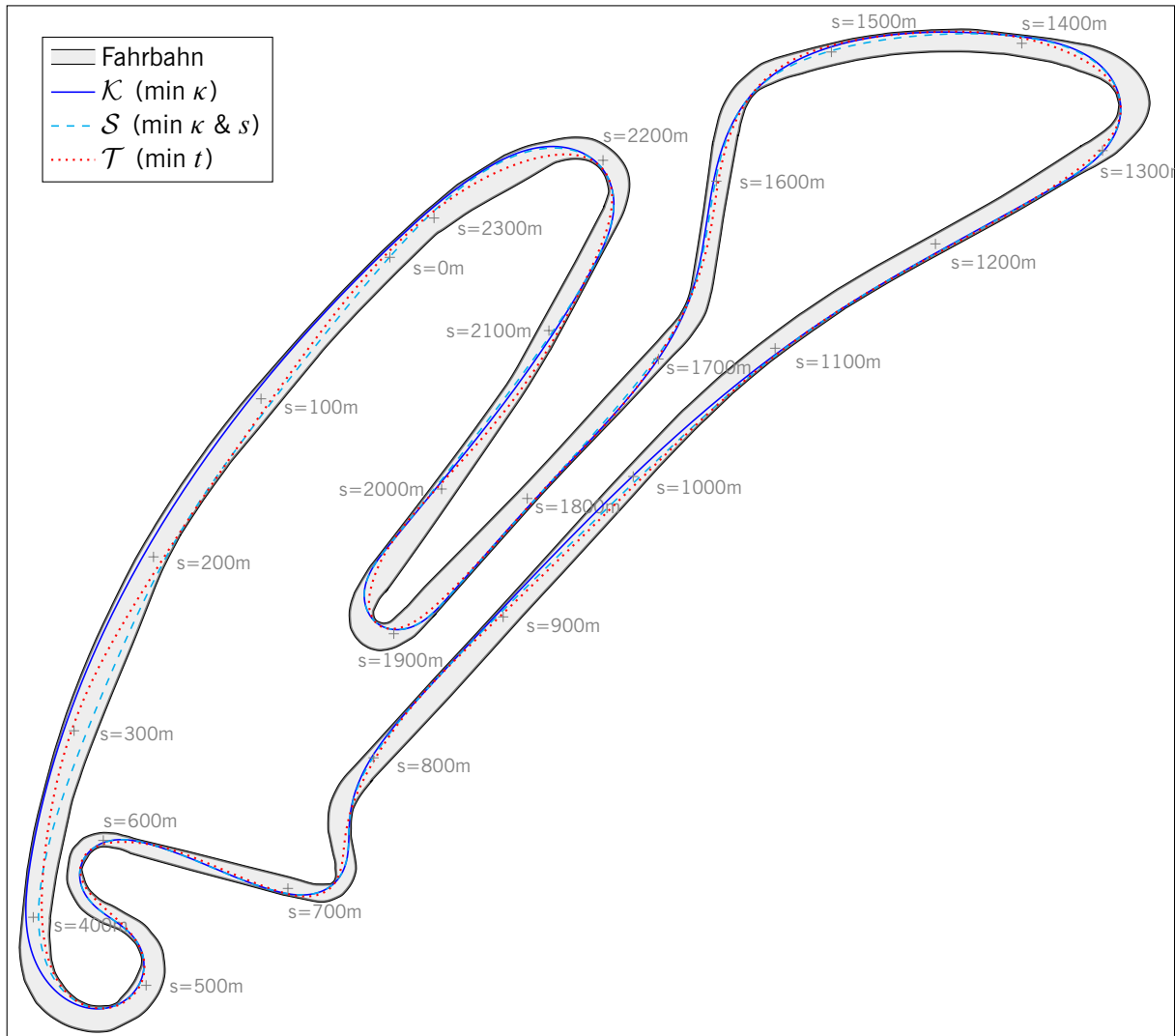


(b) Die Ausgangslinie wurde dabei um maximal 5 cm verschoben und die mittlere Verschiebung beträgt lediglich  $\overline{|e|} = 2,3$  cm (rot gestrichelte Linien).

**Bild 6.10:** Ergebnis zur Filterung einer verrauschten Linie mit den Parametern  $\lambda_1 = 0$ ,  $\lambda_2 = 1$ ,  $\lambda_3 = 5$  und  $\lambda_4 = 10$  sowie  $\lambda_e = 0,01$ .

eine – geometrisch berechnete – krümmungsminimale Linie einer – fahrdynamisch berechneten – Ideallinie nahe kommt, doch im Detail noch Abweichungen aufweist, die je nach Fahrzeug unterschiedlich ausfallen. Das Geschwindigkeitsprofil zu der krümmungsminimalen Linie wird mittels Vorwärts-/Rückwärts-Integration eines quasi-statischen Modells berechnet, wie es in Abschnitt 2.2 beschrieben ist. Darüber hinaus wird ein Vergleich zu der schon mehrfach angeführten Veröffentlichung [37] gezogen, welche eine ebenfalls krümmungsminimale Planung beschreibt und zudem erste Ergebnisse einer bis dato noch nicht veröffentlichten zeitoptimalen Planung angibt, die auf [61, 96] basiert. Um gleichzeitig zu dem Vergleich zwischen dem Kappa-Filter und der zeitoptimalen Planung auch die Ergebnisse aus [37] gegenüberstellen zu können, wird als Beispiel-Strecke der Formula E-Kurs in Berlin von 2018 für den Formula E-Wagen verwendet.<sup>3</sup> In

<sup>3</sup>Mit Dank an A. Wischnewski, A. Heilmeier und F. Christ von der TU München, die mir die Streckendaten und relevante Fahrzeugparameter zur Verfügung gestellt haben.



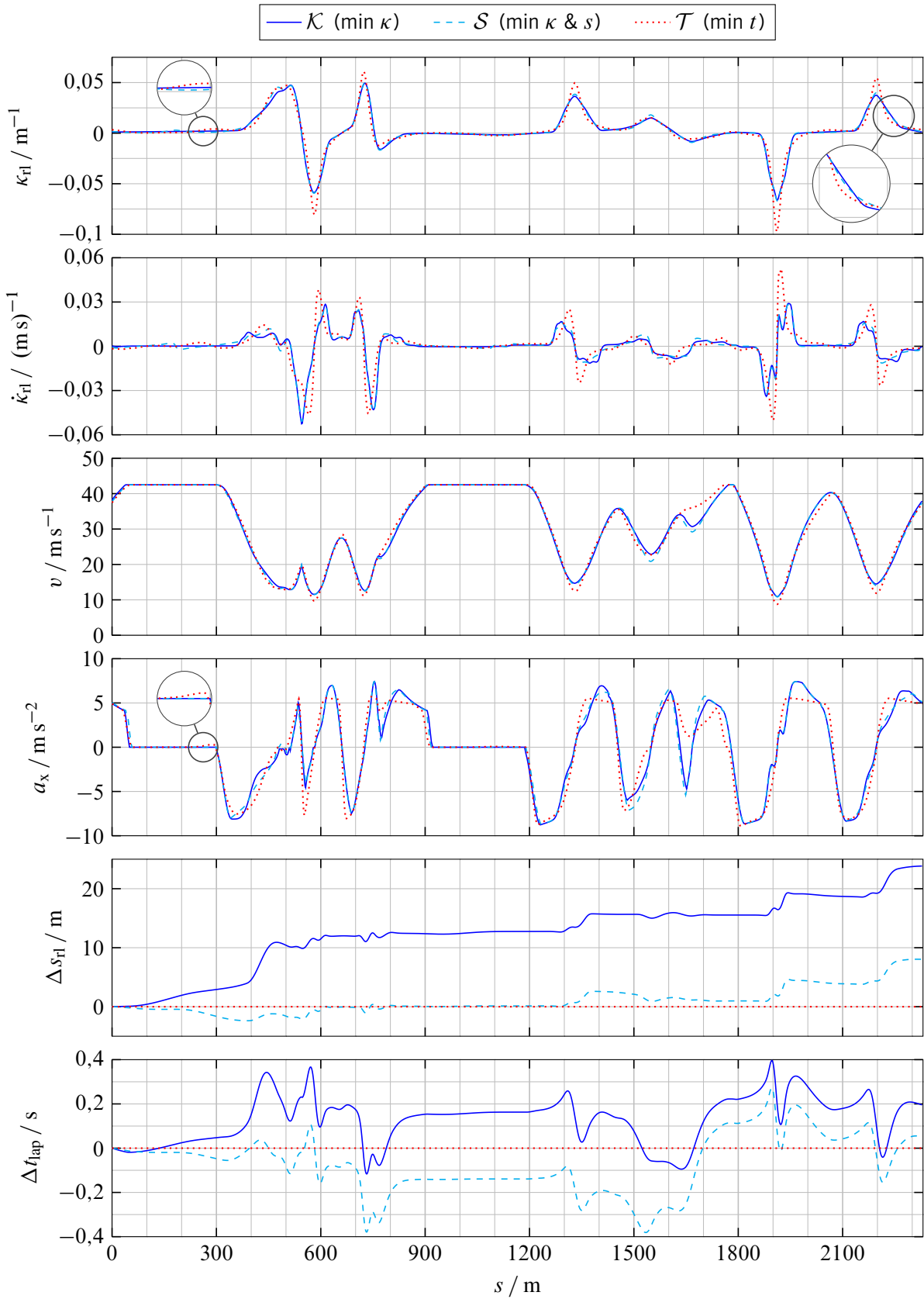
**Bild 6.11:** Vergleich der Fahrlinien krümmungsminimal  $\mathcal{K}$ , krümmungs- und wegminimal  $\mathcal{S}$  und zeitoptimal  $\mathcal{T}$  für den Formula E-Kurs in Berlin von 2018.

[37] sind Längs- und Querbefleunigung auf 80 % des physikalischen Limits begrenzt und die Maximalgeschwindigkeit beträgt  $v_{\max} = 150 \text{ km/h}$ , was auch hier berücksichtigt ist. Des Weiteren soll die Krümmung weniger als  $\kappa_{\text{rl,max}} = 0,12 \text{ m}^{-1}$  betragen.

### Krümmungsminimale vs. zeitoptimale Trajektorie

Bild 6.11 und Bild 6.12 zeigen für den gut 2,3 km langen Rundkurs folgende Trajektorien:

- $\mathcal{K}$  bezeichnet eine krümmungsminimale Linie, die mit den Parametern  $\lambda_1 = 0$ ,  $\lambda_2 = 1$ ,  $\lambda_3 = 6$  und  $\lambda_4 = 18$  sowie  $\lambda_e = 0$  berechnet wird,
- $\mathcal{S}$  ist eine Linie, die einen Kompromiss aus Krümmungs- und Wegminimierung darstellt, indem  $\lambda_1 = 0,03$  gewählt wird,
- $\mathcal{T}$  ist die zeitoptimale Linie.



**Bild 6.12:** Verlauf ausgewählter Größen zu den Trajektorien aus Bild 6.11. Die Differenzen von Weglänge  $\Delta s_{r1}$  und Fahrzeit  $\Delta t_{lap}$  beziehen sich auf die zeitoptimale Trajektorie  $\mathcal{T}$ .

Alle  $s$ -Werte beziehen sich auf die Referenzlinie. Die in Bild 6.11 eingezeichneten Fahrbahnränder berücksichtigen die Fzg-Breite von 2 m.

Bei grober Betrachtung unterscheiden sich nur die Linien in Bild 6.11 an manchen Stellen, während die Verläufe von  $\kappa$ ,  $v$  und  $a_x$  in Bild 6.12 fast gleich sind. Insgesamt ist  $\mathcal{T}$  um 0,2 s schneller als  $\mathcal{K}$  und 0,05 s schneller als  $\mathcal{S}$ , was relativ gesehen weniger als 0,23 % ausmacht. Allerdings darf die Fahrzeit niemals zur Beurteilung der Güte einer Planung herangezogen werden, solange nicht bewiesen ist, dass die Trajektorie auch tatsächlich in der Zeit abgefahren werden kann. Vielmehr dient der berechnete Zeitverlauf eher zur Analyse der Unterschiede in Trajektorien. Bild 6.12 zeigt zudem, dass der Zeitvorsprung keineswegs kontinuierlich aufgebaut wird.

Der offensichtlichste Unterschied in der Linienführung ist im Bereich  $s = 2200 \dots 470$  m zu erkennen. Wir betrachten dafür zuerst den Abschnitt  $s = 40 \dots 300$  m, in welchem alle Trajektorien dieselbe Geschwindigkeit  $v_{\max}$  haben, sodass die Zeitdifferenz ausschließlich auf der Länge der Fahrstrecke  $s_{\text{fl}}$  beruhen kann:  $\mathcal{T}$  ist  $\Delta s_{\text{fl}} \approx 3$  m kürzer als  $\mathcal{K}$ , wodurch ein Zeitvorteil von rund  $\Delta T_{\text{lap}} = \frac{\Delta s_{\text{fl}}}{v} \approx 0,07$  s entsteht. Dabei wächst der Vorsprung entgegengesetzt proportional zu  $v$ , sodass die Relevanz der Linienwahl abnimmt, je schneller man fährt (solange  $v$  weit unter dem physikalischen Limit ist). Bei  $\mathcal{S}$  ist die Weglänge in dem Abschnitt noch ca. 1,5 m kürzer als bei  $\mathcal{T}$ , wodurch  $\mathcal{S}$  ca. 0,035 s schneller als  $\mathcal{T}$  ist. Doch dieser scheinbare Vorteil von  $\mathcal{S}$  kann nur dadurch erzielt werden, dass  $\mathcal{S}$  bei  $s \geq 2200$  m, also in der Kurve davor, 3,8 m länger ist.

Eben diese Kurve bei  $s = 2200$  m erweist sich als Schlüsselstelle der Strecke: Wer hier gut herausbeschleunigt, erreicht früher die Höchstgeschwindigkeit, die für recht lange Zeit gehalten werden kann, und erzielt so einen Vorteil.  $\mathcal{T}$  fährt die Kurve extrem spitz und dreht das Fzg fast auf der Stelle, was sehr langsam geschehen muss ( $\Delta v \approx 10$  km/h). Doch das hohe Drehmoment des Elektromotors erlaubt danach eine hohe Längsbeschleunigung, die aber nur bei niedriger Querbewegung übertragen werden kann. Deswegen fällt die Krümmung hinter dem Apex schnell ab und ist im Bereich  $2220 \dots 2260$  m bei  $\mathcal{T}$  deutlich geringer als bei  $\mathcal{K}$  und  $\mathcal{S}$  (siehe Lupe in Bild 6.12). So kann  $\mathcal{T}$  schon beschleunigen, während  $\mathcal{K}$  und  $\mathcal{S}$  noch „Halbgas“ fahren müssen. Erst ab  $s \approx 2300$  m steigt  $\kappa_{\text{fl}}$  in  $\mathcal{T}$  ganz leicht an, um den Kurswinkel anzupassen, weil anderenfalls die Fahrstrecke zu lang würde. Doch diese geringe Krümmung ist problemlos, weil aufgrund der niedrigen Geschwindigkeitsbegrenzung  $v_{\max}$  genügend Kraftschlusspotential vorhanden ist.

Im Bereich  $s = 300 \dots 370$  m zeigt sich beim Anbremsen auf die Kurve ein gravierender Nachteil von  $\mathcal{K}$  und in geringerem Maße ebenso bei  $\mathcal{S}$ : Denn  $\mathcal{K}$  fährt die Kurve viel zu weit außen an, wodurch sie schon sehr früh (bei  $s \approx 400$  m) stark einlenken muss, und deswegen zu früh zu stark abbremsen muss. Man kann in Bild 6.12 den Zeitverlust deutlich erkennen. Jedoch ergibt sich dadurch ein größerer Radius für die folgende Kurve ( $s \geq 500$  m), wodurch diese schneller durchfahren werden kann, was ein Vorteil für  $\mathcal{K}$  ist. Aber da die gesamte Kurvenkombination  $s = 500 \dots 700$  m recht eng und langsam ist und daher keine so große Relevanz für das Gesamtergebnis hat, ist das kein bleibender Vorteil, sondern nur eine kurzfristige Verschiebung.

Die Kurvenkombination bei  $s = 700 \dots 800$  m ist die zweite relevante Schlüsselstelle der Strecke, um auf der folgenden Gegengeraden früh beschleunigen zu können.  $\mathcal{T}$  opfert die Kurve bei  $s = 730$  m, indem sie sie extrem eng fährt und den Apex um 3,7 m nach vorne verschiebt (sie-

he Krümmung in Bild 6.12), um den leichten Rechtsknick bei  $s = 760$  m ebenfalls früh und vor allem mit minimaler Krümmung fahren zu können, um dann so früh wie möglich beschleunigen zu können. Man sieht an  $a_x$  in Bild 6.12, wie  $\mathcal{K}$  und  $\mathcal{S}$  bei  $s \approx 770$  m fast ganz vom Gas gehen müssen, während  $\mathcal{T}$  die Beschleunigung fast ganz aufrecht erhält.

Ein Detail in der Längsbeschleunigung von  $\mathcal{T}$  fällt im Bereich vor  $s = 300$  m auf: Und zwar steigt  $a_x$  vor dem Bremspunkt leicht an (s. Lupe in Bild 6.12), obwohl die Geschwindigkeit  $v = v_{\max}$  exakt konstant bleibt. Das ist nicht etwa eine numerische Ungenauigkeit, sondern eine korrekte Abbildung des Fahrzeugverhaltens. Denn genau an dieser Stelle nimmt die Krümmung leicht zu (s. Lupe), d. h. es wird über einen Lenkwinkel  $\delta > 0$  ein Schwimmwinkel  $\beta < 0$  generiert, der zu einer Querkraft  $F_{y_r} > 0$  führt. Gemäß der DGL (3.34a) auf S. 36 des Einspurmodells führt dies zu einer Verzögerung  $\dot{v} < 0$ , wobei  $v$  in Richtung des Kurswinkels, also der Bewegungsrichtung, zeigt. Um dennoch  $v = v_{\max} = \text{const}$  halten zu können, muss eine zusätzliche Antriebskraft  $F_{x_r} > 0$  aufgebracht werden, die sich in der Längsbeschleunigung  $a_x$ , also entlang der Fzg-Längsachse, bemerkbar macht.

### Vergleich mit HEILMEIER ET AL. [37]

Die folgende Gegenüberstellung beschränkt sich auf die in [37] veröffentlichten Angaben und kann daher nicht im Detail auf Unterschiede in den geplanten Trajektorien eingehen. Stattdessen werden Unterschiede im Modell, der prinzipiellen Linienwahl sowie der Runden- und Rechenzeiten dargelegt.

Im Gegensatz zu der vorliegenden Arbeit zielt [37] auf Rennwagen ab. Daher ist in dem dortigen Modell eine geschwindigkeitsabhängige aerodynamische Abtriebskraft berücksichtigt, die in dem hiesigen Modell für Serienfahrzeuge nicht vorhanden ist. Sie bewirkt, dass bei hoher Geschwindigkeit größere Längs- und Querkräfte aufgebracht werden können, was sich natürlich sowohl auf den Verlauf der Ideallinie als auch auf die Rundenzeit auswirkt.

Während die krümmungsminimale Linie in [37, S. 20] sehr ähnlich zu Bild 6.11 ist, weist die zeitoptimale Linie in [37, S. 9] zumindest in dem dargestellten Bereich um  $s \approx 330$  m einen signifikanten Unterschied auf, indem sie dort viel weiter am rechten Rand verläuft. Möglicherweise kann das Modell in [37] aufgrund des aerodynamischen Abtriebs stärker verzögern, setzt daher den Bremspunkt später und plant deswegen, später einzulenken. Dann bleibt das Fzg bis zum Einlenkpunkt für maximale Verzögerung gerade und lässt sich daher weiter herausragen.

In [37] wird eine konstante Diskretisierung der Referenzlinie von  $\Delta s = 3$  m verwendet. Für die Algorithmen der vorliegenden Arbeit ist diese Diskretisierung für die engen Kurven zu grob, um die Fahrdynamik hinreichend exakt abzubilden. Die krümmungsminimale Planung reagiert darauf besonders empfindlich und liefert für eine 3 m-Diskretisierung zu optimistische Rundenzeiten ( $\Delta T_{\text{lap}} = 0,7$  s), aber auch die zeitoptimale Planung kann die Dynamik in den engen Kurven nicht mehr so detailliert abbilden. Daher wurden die Trajektorien im vorangegangenen Abschnitt mit einer variablen Diskretisierung  $\Delta s = 1 \dots 4$  m geplant. Tabelle 6.1 zeigt eine Gegenüberstellung der Runden- und Rechenzeiten für beide Diskretisierungen. Alle Rechenzeiten wurden mit einem

**Tabelle 6.1:** Rundenzeiten  $T_{\text{lap}}$  und Rechenzeiten  $T_{\text{cpu}}$  für den 2,3 km langen Formula E-Kurs in Berlin bei verschiedenen Methoden und Diskretisierungen  $\Delta s$ .

Methode	Fzg-Modell	$\Delta s$ in m	$T_{\text{lap}}$ in s	$T_{\text{cpu,QP}}$ in ms	$T_{\text{cpu,long}}$ in ms	$T_{\text{cpu,tot}}$ in s
[37] min $\kappa$	GGV-Diagramm	3	86,13	$4 \times 850$	65	18
[37] min $t$	Punktmasse	3	?	–	–	37
[37] min $t$	Zweispur	3	84,90	–	–	151
$\mathcal{K}$	GG-Diagramm	3	85,53	$4 \times 16$	18	0,116
$\mathcal{S}$	GG-Diagramm	3	85,52	$4 \times 16$	16	0,105
$\mathcal{T}$	Einspur	3	85,97	–	–	1,5
$\mathcal{K}$	GG-Diagramm	1 ... 4	86,24	$4 \times 19$	15	0,130
$\mathcal{S}$	GG-Diagramm	1 ... 4	86,10	$4 \times 19$	15	0,119
$\mathcal{T}$	Einspur	1 ... 4	86,05	–	–	1,9

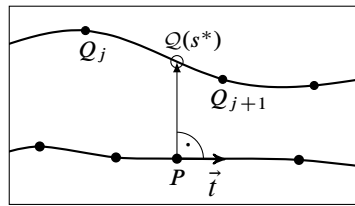
Notebook mit einer INTEL Core i7-CPU ermittelt, wobei in [37] PYTHON und hier MATLAB zum Einsatz kommt. Bei den Berechnungen mit separater Quer- und Längsplanung zeigt die Tabelle neben der Gesamtrechenzeit  $T_{\text{cpu,tot}}$  auch die reine Optimierungszeit für das Lösen des QPs der Querplanung  $T_{\text{cpu,QP}}$  sowie die Zeit für die Längsplanung  $T_{\text{cpu,long}}$ . Die Gesamtzeit beinhaltet einen gewissen Overhead, der u. a. das Aufstellen des QPs und die Spline-Interpolationen beinhaltet.

Die Rundenzeiten sind bei allen verglichenen Methoden ähnlich. Die krümmungsminimale Linie lässt sich viel schneller berechnen als die zeitoptimale. Insgesamt sind alle Rechenzeiten der Methoden aus der vorliegenden Arbeit deutlich kürzer als in [37].

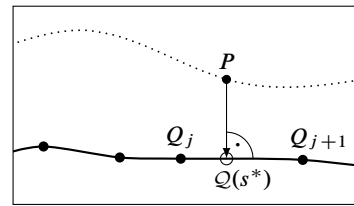
## Fazit

Als Fazit für die Fragestellung, wie gut eine krümmungsminimale Linie zur Approximation einer Ideallinie geeignet ist, lässt sich festhalten:

- Die Linienwahl ist i. A. gut, weil sie zu einer Rennlinie führt, mit der ähnliche Rundenzeiten wie mit der (zeitoptimalen) Ideallinie erreichbar sind.
- In vielen Situationen kann eine bessere Rennlinie erzielt werden, wenn zusätzlich zur Krümmung auch die Weglänge berücksichtigt wird. Die optimale Gewichtung für Krümmung und Weglänge ist jedoch strecken- und fahrzeugspezifisch und kann nur empirisch ermittelt werden (vgl. auch [12, S. 1510]).
- Die krümmungsminimale Linie berücksichtigt weder fahrzeug- noch streckenspezifische Eigenschaften, d. h. man erhält bspw. für unterschiedliche Motorisierungen oder für jede Fahrtrichtung dieselbe Linie. Vorteile durch bspw. frühes Herausbeschleunigen auf eine lange Gerade können nicht genutzt werden.



(a)  $P$  liegt auf dem Spline, zu dem das Lot gefällt wird.



(b) Match-Punkt  $Q(s^*)$  und Lot ergeben sich erst durch das Matching.

**Bild 6.13:** Unterschiedliche Zielsetzungen zum Matchen eines Punkts  $P$  auf einen Spline  $Q$ .

- Die Modellierung als Massepunkt berücksichtigt nur die Beschleunigung im Schwerpunkt. Dabei kann es passieren, dass diese innerhalb des Kammschen Kreises liegt, obwohl die Reifenkräfte an einer der beiden Achsen außerhalb liegen, sodass eine scheinbar fahrbare Trajektorie das Limit überschreitet (vgl. auch den *Herborn-Unfall* von 1987 [15, S. 1 f.]). Bei Rennwagen ist die Gewichtsverteilung weitgehend gleich, wodurch eine Approximation als Massepunkt ganz gut möglich ist, während bei Serienfahrzeugen die Vorderachse meist deutlich stärker belastet ist.
- Die Rechenzeiten der krümmungsminimalen Linie sind deutlich niedriger.

## 6.3 Matching-Algorithmus zur Referenzierung eines Punktes auf einem Spline

In der vorliegenden Arbeit bezeichnet *Matching* den Vorgang, bei welchem auf einem kubischen Spline  $\vec{Q} : \mathbb{R} \rightarrow \mathbb{R}^2$  ein Punkt  $\vec{Q}(s^*)$  derart ermittelt wird, dass eine Gerade durch diesen Punkt gelegt werden kann, welche sowohl durch einen weiteren vorgegebenen Punkt  $P$  verläuft als auch orthogonal entweder zu  $\vec{Q}$  selbst oder einem anderen Spline ist. So gibt es je nach Orthogonalitätsbedingung zwei Arten des Matchings, die in Bild 6.13 dargestellt sind:

- In Bild 6.13(a) liegt  $P$  auf einem Spline und gesucht ist der Punkt  $\vec{Q}(s^*)$ , durch den das Lot verläuft. Dieser Fall tritt auf, wenn zu einem Punkt  $P$  auf der Referenz der Abstand zum Fahrbahnrand  $\vec{Q}$  bestimmt werden soll.
- In Bild 6.13(b) ist  $P$  ein beliebiger Punkt, der nicht auf einem definierten Spline liegen muss. Gesucht ist die Projektion von  $P$  auf  $\vec{Q}$ , sodass die Verbindungslinie orthogonal zu  $\vec{Q}(s^*)$  ist. Dieser Fall tritt auf, wenn die Ego-Position  $P$  auf die Referenz  $\vec{Q}$  projiziert wird.

Unabhängig davon, welche Linie die Referenz oder der Fahrbahnrand ist, wird im Folgenden der gegebene Punkt mit  $P$  und der gesuchte Match-Punkt mit  $Q^*$  bzw.  $\vec{Q}(s^*)$  bezeichnet. Die Situation (a) ist einfacher, weil der Normalenvektor dort konstant ist, während er in (b) von der  $s$ -Koordinate abhängt. Da das Vorgehen aber in beiden Fällen ähnlich ist, wird vorrangig der Fall (b) behandelt.

Das Matching besteht aus 2 Schritten: Zuerst wird in Abschnitt 6.3.1 bestimmt, welches Spline-Segment von  $Q_j$  bis  $Q_{j+1}$  für ein orthogonales Matching verwendet werden kann. Anschließend beschreibt Abschnitt 6.3.2 das eigentliche Matching auf dieses Segment.

### 6.3.1 Spline-Zell-Suche (cell search)

Wenn es in dem  $j$ -ten Spline-Segment (also zwischen den Stützstellen bei  $Q_j$  und  $Q_{j+1}$ ) einen Punkt  $\vec{Q}(s^*)$ ,  $s^* \in [s_j, s_{j+1})$ , gibt, an dem der Verbindungsvektor von  $P$  zu  $\vec{Q}(s^*)$  in einem rechten Winkel zum Spline  $\vec{Q}$  steht, muss der Verbindungsvektor von  $P$  zu  $\vec{Q}(s_j)$  im rechten oder stumpfen Winkel stehen und von  $P$  zu  $\vec{Q}(s_{j+1})$  im spitzen. Kandidaten für geeignete Segmente können also mit dem Tangentenvektor

$$\vec{t}(s) = \frac{d}{ds} \vec{Q}(s) \quad (6.39)$$

durch die Indexmenge

$$\mathbb{K} = \{j \mid \vec{t}(s_j) \cdot (\vec{q}(s_j) - \vec{p}) \leq 0 \wedge \vec{t}(s_{j+1}) \cdot (\vec{q}(s_{j+1}) - \vec{p}) > 0\} \quad (6.40)$$

beschrieben werden, wobei der Tangentenvektor  $\vec{t}$  nur an den Stützstellen benötigt wird und daher ohne Rechnung aus den Spline-Koeffizienten extrahiert werden kann.

Gibt es mehrere Kandidaten, wird derjenige mit dem geringsten Abstand zu  $P$  verwendet. Um nicht alle Stützstellen der Referenz auswerten zu müssen, kann die Suche nach einem Kandidaten auch gleich an der Stützstelle beginnen, die am nächsten an  $P$  liegt. In den meisten Fällen stellt eine der beiden dichtesten Stützstellen bereits einen Kandidaten gemäß (6.40) dar. Anderenfalls gibt das Vorzeichen des Skalarprodukts die Suchrichtung an. Eine andere Strategie startet bei dem zuletzt gefundenen Match-Punkt und sucht den nächsten Kandidaten in positiver Fahrtrichtung. Dabei muss aber sicher gestellt sein, dass der kartierte Fahrbahnabschnitt in der Zwischenzeit nicht verlassen wurde, weil der Algorithmus ansonsten am falschen Abschnitt „kleben bleibt“.

Oftmals will man nur Abschnitte matchen, die näherungsweise dieselbe Orientierung haben, was dann bei der Wahl der Kandidaten berücksichtigt werden muss. Im Fall (a) bedeutet dies, dass Referenz und Rand ähnlich orientiert sein müssen, und für (b), dass die Fahrzeugausrichtung, also der Gierwinkel, zur Richtung der Referenz passt. Nehmen wir als Beispiel, dass in Bild 6.7(a) auf S. 108 der Punkt  $P$  auf ein  $L_j$  gematcht werden soll. Ohne Berücksichtigung der Orientierung käme  $L_3$  heraus, mit jedoch  $L_1$ . Die Orientierungsbedingung kann auch mit einer Abstandsbedingung verknüpft werden: „Die Orientierung darf nur dann vernachlässigt werden, wenn andererseits der resultierende Punkt weiter als ... m entfernt ist.“ So verhindert man, dass aufgrund stark unterschiedlicher Orientierung anstelle eines naheliegenden Abschnitts ein falscher, weit entfernter Streckenabschnitt verwendet wird, der zufällig in der Orientierung passt.

Schließlich kann mit einer dieser Strategien ein Spline-Segment ermittelt werden, für welches das eigentliche Matching, also die Bestimmung von  $\vec{Q}(s^*)$  durchgeführt werden kann.



### 6.3.2 Analytische Bestimmung des Match-Punkts

In dem gesuchten Punkt  $\vec{Q}(s^*)$  steht der Verbindungsvektor orthogonal auf dem Tangentenvektor (6.39), sodass das Skalarprodukt

$$\left(\vec{Q}(s^*) - \vec{p}\right) \cdot \left(\frac{d}{ds}\vec{Q}(s^*)\right) = 0, \quad s^* \in [s_j, s_{j+1}), \quad (6.41)$$

ist. Mit  $\vec{Q}(s) = [\mathcal{Q}_x(s) \quad \mathcal{Q}_y(s)]^\top$  und der Schreibweise  $\vec{Q}_j(s)$  für das Polynom des  $j$ -ten Segments lässt sich (6.41) als

$$0 = \underbrace{(\mathcal{Q}_{j,x}(s^*) - p_x) \mathcal{Q}'_{j,x}(s^*)}_{\mathcal{R}_{j,x}(s^*)} + \underbrace{(\mathcal{Q}_{j,y}(s^*) - p_y) \mathcal{Q}'_{j,y}(s^*)}_{\mathcal{R}_{j,y}(s^*)} \quad (6.42)$$

schreiben. Für  $x$  und  $y$  liegt  $\mathcal{Q}_j$  jeweils als Polynom in der Form

$$\mathcal{Q}_j(s) = c_3 \tilde{s}^3 + c_2 \tilde{s}^2 + c_1 \tilde{s} + c_0, \quad \tilde{s} := s - s_j \in [0, s_{j+1} - s_j), \quad (6.43)$$

vor. Damit ergibt sich

$$\begin{aligned} \mathcal{R}_j(s) = & 3c_3^2 \tilde{s}^5 + 5c_2c_3 \tilde{s}^4 + (2c_2^2 + 4c_1c_3) \tilde{s}^3 + 3(c_1c_2 + c_3(c_0 - p)) \tilde{s}^2 + \dots \\ & + (c_1^2 + 2c_2(c_0 - p)) \tilde{s} + c_1(c_0 - p). \end{aligned} \quad (6.44)$$

So erhält man in (6.42) also die Summe aus zwei Polynomen 5. Ordnung. Eine der Nullstellen des resultierenden Polynoms ist die Lösung von (6.41), die anderen vier sind komplex oder liegen außerhalb des Spline-Segments. In Ausnahmefällen kann es passieren, dass die führenden Koeffizienten null sind, sodass sich die Ordnung reduziert, was bei der Nullstellenberechnung natürlich zu berücksichtigen ist. Falls  $[c_{x,0} \quad c_{y,0}]^\top = \vec{p}$  ist, sodass  $\vec{\mathcal{R}}_j(s)$  keinen konstanten Anteil besitzt, liegt  $P$  auf der Stützstelle  $\mathcal{Q}_j$ , sodass  $\tilde{s} = 0$  bzw.  $s^* = s_j$  die Lösung von (6.41) ist.

Bei der Berechnung der Fahrbahnbreite gemäß Fall (a) ist der Tangentenvektor konstant und muss nicht über  $\vec{Q}'(s)$  in die Gleichung eingehen, sodass sich in (6.44) nur ein Polynom dritter anstelle fünfter Ordnung ergibt.

Die Nullstellenberechnung eines Polynoms  $n$ -ter Ordnung lässt sich auf die Eigenwertberechnung einer  $n \times n$ -Matrix zurückführen, indem die Ursprungsgleichung zuerst in die Form

$$0 = s^n + \sum_{k=0}^{n-1} r_k s^k \quad (6.45)$$

gebracht wird und dann aus den Koeffizienten  $r_k$  eine Matrix  $\mathbf{A}$  aufgestellt wird, deren charakteristisches Polynom dieselben Nullstellen wie (6.45) hat. Eine mögliche Matrix ergibt sich aus der *Regelungsnormalform* einer Zustandsgleichung [29, S. 250]:

$$0 = s^n + \sum_{k=0}^{n-1} r_k s^k \Leftrightarrow 0 = \det(s\mathbf{I}_n - \mathbf{A}), \quad \mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ & & & \ddots & \\ & & & & 1 \\ -r_0 & -r_1 & -r_2 & \dots & -r_{n-1} \end{bmatrix} \quad (6.46)$$

Die Eigenwerte von  $\mathbf{A}$  lassen sich bspw. mit dem „Linear Algebra Package“ LAPACK, einer in Fortran programmierten Bibliothek, berechnen, welche Funktionen der elementaren linearen Algebra zur Verfügung stellt. LAPACK verwendet eine Schnittstelle zu BLAS, um die dort meist hocheffizient realisierten Basis-Funktionen zu nutzen. Es ist auch Bestandteil der Intel MKL [44].

Aus dem so ermittelten Match-Punkt  $Q(s^*)$  ergibt sich auch die Distanz  $d$  zwischen  $Q(s^*)$  und  $P$ . Das Vorzeichen von  $d$  ist

$$\text{sgn}(d) = \begin{cases} \text{sgn} \left( \vec{e}_3 \cdot (\vec{t}^* \times (\vec{q}^* - \vec{p})) \right), & \text{Fall (a),} \\ \text{sgn} \left( \vec{e}_3 \cdot ((\vec{q}^* - \vec{p}) \times \vec{t}^*) \right), & \text{Fall (b),} \end{cases} \quad (6.47)$$

wobei  $\vec{q}^*$  der Ortsvektor zum Match-Punkt  $Q(s^*)$  ist und  $\vec{t}^*$  der Tangentenvektor nach (6.39) am Match-Punkt.

Ein globales Matching – d. h. die Zellsuche wird auf die gesamte Strecke ausgeweitet und nicht lokal begrenzt – dauert für eine 4,5 km lange Strecke auf einer Intel Core i7 Mobile-CPU ca. 60 – 70  $\mu\text{s}$  (je nachdem ob mit oder ohne Berücksichtigung des Orientierungswinkels). Davon entfällt der größte Teil mit ca. 40  $\mu\text{s}$  auf die Eigenwertberechnung.

## 7 Zyklische Echtzeit-Planung mit Objekten

In den vorangegangenen Kapiteln lag der Fokus auf der Trajektorienplanung für die gesamte Strecke, welche offline berechnet wird. In Hinblick auf den realen Straßenverkehr müssen sich verändernde Umgebungsbedingungen, wie Hindernisse, einkalkuliert werden, was eine Neuplanung während der Fahrt erfordert. Da die hierfür verfügbare Rechenzeit i. A. nicht ausreicht, um die gesamte Strecke neu zu planen, benötigt man eine *zyklische Online-Planung*, die in jedem Zyklus eine Trajektorie für den kommenden Abschnitt plant. Sie wird auch als Echtzeit-Planung bezeichnet, was in Abschnitt 7.1 erklärt ist. Anschließend sind in Abschnitt 7.2 die Auswirkungen einer abschnittswisen Planung auf das Optimierungsproblem beschrieben. In Abschnitt 7.3 ist die Einbindung statischer Objekte erklärt und Abschnitt 7.4 präsentiert abschließend Simulations- und Messergebnisse der Echtzeit-Planung.

### 7.1 Echtzeit- und Rechenzeitanforderungen

Ein Echtzeitsystem garantiert die Berechnung seiner Ergebnisse innerhalb eines festgelegten Zeitintervalls, dessen Länge von der Anwendung abhängt [66, S. 84]. Eine Echtzeitanforderung wird als *hart* bezeichnet, wenn sie nachweislich und strikt eingehalten werden muss und das Ergebnis bei einer Überschreitung ungültig ist. Ohne den Nachweis einer strikten Einhaltung wird sie als *weich* bezeichnet. Harte Echtzeitanforderungen dürfen also nicht mit der Ausführungsgeschwindigkeit verwechselt oder gar gleichgesetzt werden [72, Kap. 2.4].

Bei einer numerischen nichtlinearen Optimierung ist es prinzipiell unmöglich, nach einer bestimmten Zeit eine Lösung zu garantieren, da ihre Konvergenz weder garantiert noch im Vorfeld bekannt ist. Empirisch lässt sich jedoch eine typische Ausführungszeit (Mittelwert, Quantile etc.) ermitteln. Sie ist normalerweise sehr viel kürzer als die zeitliche Länge der geplanten Trajektorie, sodass für den *gesamten Planer eine weiche Echtzeitanforderung* genügt. Jedoch ist eine möglichst kurze Rechenzeit viel wichtiger als eine garantierte, denn je kürzer sie ist, umso eher kann einem plötzlich auftretenden Hindernis ausgewichen werden.

Allerdings muss beachtet werden, dass das Fzg während der Planungszeit entlang der letzten Trajektorie weiterfährt. Erst nach Abschluss der Neuplanung kann eine veränderte Trajektorie an den Regler übertragen werden. Daher muss die neue Trajektorie bis zu diesem Zeitpunkt völlig identisch zu der vorherigen sein, da sie ansonsten rückwirkend verändert werden müsste, was natürlich unmöglich ist. Hinzu kommen noch Totzeiten in den Steuergeräten, die v. a. bei Fahrzeugen mit Serienfreigabe nicht zu vernachlässigen sind. Um den Punkt zu bestimmen, ab dem eine Veränderung möglich ist, muss die benötigte Rechenzeit präzisiert werden. Ist die Prädiktion zu pessimistisch, wird die Planung unnötig träge und kann im schlimmsten Fall eine vermeidbare Kollision nicht mehr verhindern. Ist sie zu optimistisch, liegt die Trajektorie zu spät vor, wodurch sie ggf.

„unfahrbar“ und daher ungültig ist. Durch die Festlegung des Zeitlimits wird aus der ehemals weichen eine *harte Echtzeitanforderung*. Da das Zeitlimit durch eine interne Prädiktion innerhalb gegebener Schranken adaptiv bestimmt wird, variiert die Zykluszeit des Planers. Betrachtet man den Planer als Blackbox, kann man daher nur eine weiche Echtzeitanforderung vorgeben.

Darüber hinaus ist eine Sicherheitsstrategie wichtig, die das Fzg im Falle einer inakzeptablen Zeitüberschreitung entlang einer sicheren Trajektorie in einen sicheren Zustand überführt. Sie ist jedoch nicht Gegenstand dieser Arbeit. Im Rahmen dieses Forschungsprojekts besteht die Strategie aus einer schnellstmöglichen Verzögerung in den Stillstand. Sie kann entlang einer *krümmungsminimalen Linie* innerhalb der Streckenränder erfolgen, welche mit dem *Kappa-Filter* aus Abschnitt 6.2 in kurzer Zeit und mit hoher Konvergenz (weil QP) berechnet werden kann.

## 7.2 Optimierungsproblem bei gleitendem Horizont

Bei der Planung für einen Rundkurs der Länge  $S_{RK}$  liegt ein periodisches Optimierungsproblem (7.1a) vor, weil  $\mathbf{x}(s=0) \stackrel{!}{=} \mathbf{x}(s=S_{RK})$  gefordert wird. Es ist vergleichbar mit einem Problem mit unendlich langem Planungshorizont, bei dem sich der Streckenverlauf periodisch wiederholt, und von dem man aus der Mitte eine Runde extrahiert. Dieser Horizont wird bei der Echtzeit-Planung auf einen Abschnitt  $S_{Ab} < S_{RK}$  reduziert, wodurch sich ein völlig neues Optimierungsproblem (7.1b) ergibt (o. B. d. A. beginne der Abschnitt hier bei  $s = 0$ ).

$$\mathbf{x}_{RK} = \arg \min_{\mathbf{x}} \int_0^{S_{RK}} \Phi(\mathbf{x}) ds \quad \text{u. B. v.} \quad \mathbf{x}(0) \stackrel{!}{=} \mathbf{x}(S_{RK}), \quad (7.1a)$$

$$\mathbf{x}_{Ab} = \arg \min_{\mathbf{x}} \int_0^{S_{Ab}} \Phi(\mathbf{x}) ds + \vartheta(\mathbf{x}(S_{Ab})) \quad (7.1b)$$

Bei identischen Voraussetzungen ohne Objekte sollten beide Trajektorien auf dem Abschnitt identisch sein, woraus sich eine Bedingung für den MAYER-Term  $\vartheta(\mathbf{x}(S_{Ab}))$  der Endkosten ableitet:

$$\forall s \in [0, S_{Ab}] : \mathbf{x}_{RK} = \mathbf{x}_{Ab} \quad \Rightarrow \quad \vartheta(\mathbf{x}(S_{Ab})) = \int_{S_{Ab}}^{S_{RK}} \Phi(\mathbf{x}) ds, \quad \mathbf{x}(S_{RK}) \stackrel{!}{=} \mathbf{x}(0) \quad (7.2)$$

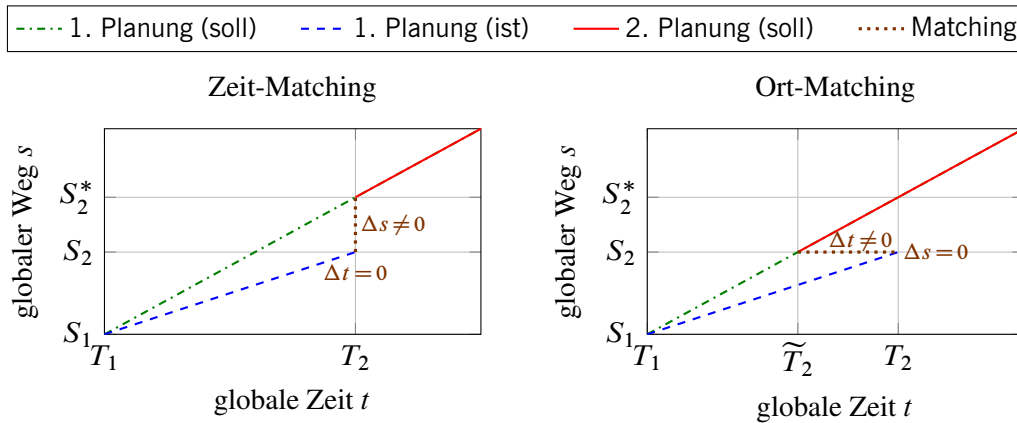
Leider lässt sich (7.2) nicht angeben, ohne eben die restliche Strecke zu berechnen. Würde man  $\vartheta \equiv 0$  setzen, würde die zeitoptimale Lösung bei Kurven am Ende stets mit maximaler Geschwindigkeit geradeaus in den Fahrbahnrand fahren. Bei einer Neuplanung im Grenzbereich, also ohne physikalische Reserven, braucht man die Garantie, dass im nächsten Planungsschritt eine physikalisch mögliche Anschlusstrajektorie existiert. Diese Bedingung ist im realen Straßenverkehr zwar genauso relevant, aber leichter zu erfüllen, weil sich i. d. R. fahrdynamisch sichere Zustände (z. B. Fahrstreifenmitte mit vorgegebener Geschwindigkeit) angeben lassen. In Hinblick auf die Zeitoptimalität würde eine solche Endbedingung jedoch die Güte der Trajektorie extrem mindern. Hier

wird nun gefordert, dass die Trajektorie am Ende des Planungshorizonts möglichst nahe an der Offline-Lösung enden muss, indem die Abweichung dazu bestraft wird. Die Geschwindigkeit der Offline-Lösung muss nicht erreicht werden, sie darf nur nicht überschritten werden. Auch wenn dies aufgrund von Abweichungen im Endpunkt de facto keine Garantie für die Existenz einer Anschlusstrajektorie liefert, kann man für alle praktischen Fälle davon ausgehen. Denn bei ausreichendem Planungshorizont sind die Auswirkungen der Rückführung auf die Offline-Lösung gering, weil immer nur der Anfang jeder Trajektorie tatsächlich verwendet wird. Und der Planungshorizont wird immer so groß gewählt, wie es die Rechenzeit erlaubt, um den Einfluss des Endpunkts zu minimieren (→ Bild 7.2).

Wenn die Offline-Planung, aus welcher  $\vartheta(\mathbf{x}(S_{Ab}))$  stammt, dieselben Parameter verwendet hat, wie die Online-Planung, ergeben sich weitgehend identische Lösungen. Wenn der Online-Planung jedoch abweichende Parameter (z. B. Fzg-Masse oder Kraftschlussbeiwert) übergeben werden, ergibt sich mathematisch ein neues Optimierungsproblem. Sein Optimum wird dann i. A. weder der zeitoptimalen Lösung für den einen Parametersatz noch für den anderen entsprechen, sondern irgendeine andere Trajektorie bilden, die allerdings weiterhin die Modellgleichung erfüllt.

Die abschnittsweise Planung verfügt ohne den MAYER-Term  $\vartheta$  ausschließlich über Informationen über den Streckenverlauf innerhalb ihres Planungs- oder auch Vorausschauhorizonts. Dass für die Berechnung der Ideallinie Kenntnis über die gesamte Ideallinie in Form jenes MAYER-Terms unbedingt notwendig ist, kann man sich anhand eines menschlichen Rennfahrers veranschaulichen: In seiner ersten Runde auf einer ihm noch unbekannten Strecke fährt er niemals seine persönliche Bestzeit. Wie gut seine Linie ist, hängt davon ab, wie weit er nach vorne schauen kann. Erst wenn er den Streckenverlauf kennt, weiß er aus dem Gedächtnis, wie er hinter dem Abschnitt, den er aktuell sieht, weiterfahren kann, und passt seine Trajektorie entsprechend an. Dieses „Gedächtnis“ ist der Speicher, in dem die Offline-Lösung liegt, welche über den MAYER-Term die Information über den prinzipiellen Verlauf hinter dem Planungshorizont gibt. Ohne diese Information kann die Planung genauso wie der Rennfahrer je nach gewähltem MAYER-Term bzw. prädiziertem Verlauf mehr oder weniger gute Trajektorien ermitteln, aber i. A. nicht das Optimum finden. Liegen lokal veränderte Voraussetzungen, wie ein Hindernis, vor, gilt die Annahme, dass ihre Auswirkung lokal begrenzt ist, und den weiteren Verlauf nicht wesentlich beeinflusst. Ist die Änderung permanent, wie eine veränderte Massenverteilung oder ein anderes Reifenverhalten aufgrund der Abnutzung, führt die Planung eine vollständige Neuberechnung des Rundkurses durch (dies kann im Hintergrund ausgeführt werden) und passt so den MAYER-Term an. Ein professioneller menschlicher Fahrer muss aufgrund seiner Erfahrung und seiner Intelligenz die Auswirkung prognostizieren und seine Trajektorie adaptieren.

Neben der Wahl des Endpunkts  $\mathbf{x}(S_{Ab})$  ist die des Anfangspunkts  $\mathbf{x}(0)$  wichtig, welcher in Form einer NB (vgl. (4.4b) auf S. 49) fest vorgegeben wird. Denn er beeinflusst maßgeblich, wie die am Regler ankommende resultierende Trajektorie aussieht, die sich aus der Aneinanderreihung der jeweiligen Trajektorienstücke aller Planungsschritte ergibt. Wenn die Einzeltrajektorien am Übergang von einem zum nächsten Planungsschritt nicht stetig differenzierbar oder gar unstetig ineinander übergehen, ist auch die abzufahrende Solltrajektorie nicht mehr stetig (differenzierbar). Dass eine unstetige Trajektorie physikalisch nicht fahrbar und schon gar nicht angenehm fahrbar



**Bild 7.1:** Bei der 1. Planung ist die Ist-Geschwindigkeit niedriger als ihr (konstanter) Sollwert. Zum Zeitpunkt  $T_2$  befindet sich das Fzg am Ort  $S_2$ , wo neu geplant wird. Die Neuplanung kann entweder an diesem Zeitpunkt (links) oder an diesem Ortspunkt (rechts) beginnen.

wäre, liegt auf der Hand. Um also entsprechend stetige Trajektorien zu generieren, muss die zu planende Trajektorie auf dem *entsprechenden Punkt* der letzten Soll-Trajektorie beginnen (auf die Alternative, den Ist-Zustand zu nehmen, wird später noch eingegangen). Doch was genau ist der „entsprechende Punkt“? – Ist es ein Zeitpunkt oder ein Ortspunkt? Beide Größen sind monoton wachsend. Im Folgenden wird besprochen, welcher Punkt die bessere Wahl ist.

Solange keine Regelabweichung vorliegt, sind beide Strategien qualitativ gleich, sofern Planung und Regler dieselbe Strategie verwenden. Doch in der Realität kommt es zu Regelabweichungen, wie Bild 7.1 schematisch zeigt: Zum Zeitpunkt  $T_2$  soll eine Neuplanung erfolgen. Das Fzg befindet sich aufgrund einer Geschwindigkeitsabweichung allerdings nicht wie geplant an  $S_2^*$ , sondern an  $S_2$ . Beim Zeit-Matching (links) betrachtet man den aktuellen Zeitpunkt ( $T_2$ ) und setzt die Trajektorie wie zuvor geplant zu diesem Zeitpunkt und räumlich vor dem Fzg stetig fort. Die Regelabweichung  $\Delta s = S_2^* - S_2$  bleibt also im nächsten Abschnitt bestehen. Beim Ort-Matching (rechts) projiziert man die aktuelle Fzg-Position auf die geplante Linie (vgl. Matching-Algorithmus in Abschnitt 6.3) und startet beim dortigen Zustand, wodurch jede Neuplanung am aktuellen Ort beginnt und der Zeitverlauf einen Sprung  $\Delta t = \tilde{T}_2 - T_2$  aufweist.

Welche der beiden Methoden ist näher am menschlichen Verhalten? – Zwei Beispiele:

- Auf einem Streckenabschnitt hält ein Fahrer eine Geschwindigkeit von 60 km/h für optimal. Weil er abgelenkt ist, fährt er nur 50 km/h und erreicht den nächsten Abschnitt 15 s später. Wird er dort 10 km/h schneller fahren als normal, um die Zeitdifferenz aufzuholen, obwohl er die höhere Geschwindigkeit für nicht angemessen hält? In den meisten Fällen wohl nicht, weil sich der Mensch intuitiv am Ort orientiert und keine exakte „innere Uhr“ hat.
- Betrachten wir ein Navi als Pendant zum Planer und den Fahrer zum Regler. Würde das Navi sich auf die vorgesehen Fahrzeit matchen, käme es zu bizarren Situationen: Führe man schneller als geplant, würde es noch die Anweisung zur vergangenen Kreuzung geben. Müsste man unerwartet anhalten, würde es trotzdem mit den Anweisungen fortfahren.

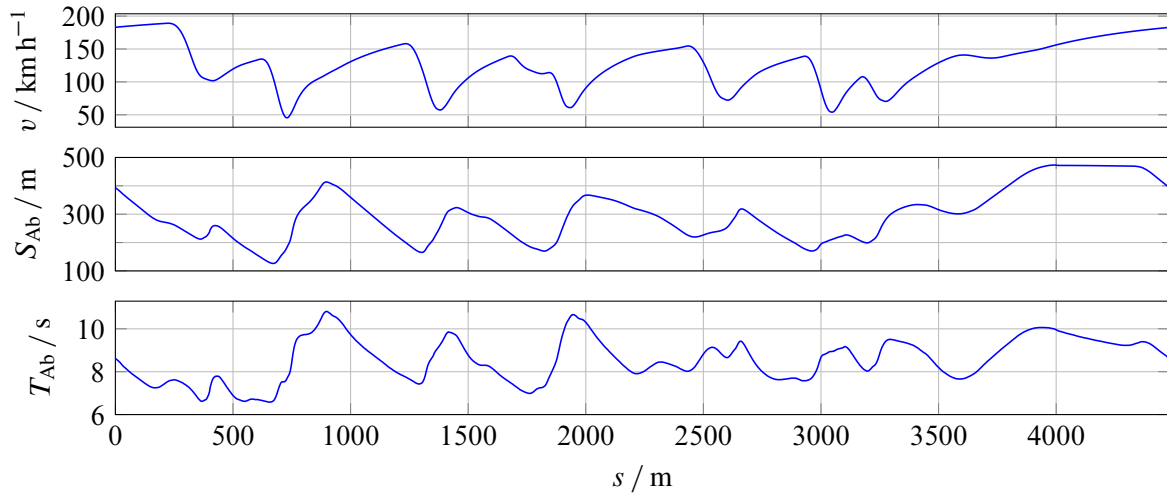
Beide Beispiele zeigen, dass ein Ort-Matching im Straßenverkehr realistischer ist als ein Zeit-Matching. Zudem ist es im Grenzbereich unmöglich, verlorene Zeit aufzuholen, weil man die fahrdynamische Grenze überschreiten würde. Allerdings kann ein Ort-Matching kein Stillstands-Szenario abbilden. Doch tritt es fast nie auf, dass man eine vorgegebene Zeitspanne im Stillstand warten möchte. Eher wird die Zeit durch äußere Ereignisse, wie eine Lichtsignalanlage, bestimmt.

Die zyklische Planung startet also immer an dem örtlich korrespondierenden *Soll*-zustand der letzten Trajektorie. Anders formuliert wird für den unabhängigen Parameter  $s$ , also der Referenz-Längsrichtung, der Istzustand verwendet, während alle anderen Zustände aus den Sollgrößen stammen. Hingegen würde bei einer klassischen MPC am vollständigen *Ist*-zustand gestartet, um diesen stetig fortzusetzen. Dabei würde die Regelabweichung in jedem Planungsschritt implizit auf null „zurückgesetzt“, sodass zum einen der Stellgrößenverlauf unstetig wäre und zum anderen etwaige unterlagerte Regelkreise nur die Abweichung registrieren könnten, die sich zwischen zwei Planungsschritten aufbaut. Das hätte zur Folge, dass konstante Störungen (evtl. trotz I-Anteil im Regelkreis) nur langsam ausgeregelt würden. Die resultierende Trajektorie verlief dann im Zustandsraum oft neben der eigentlichen Planung und würde erst, wenn die MPC aufgrund einer inakzeptablen Abweichung dann doch stark gegensteuert, abrupt in den gewünschten Zustand schwenken. Da ein solcher Verlauf an die Bahn eines wildernden Hundes erinnert, wird sie auch als *Hundekurve* bezeichnet, siehe [89, S. 6]. Das Hundekurvenproblem kann bei dem hier verwendeten Matching nicht auftreten, weil nur der Zeitfehler zurückgesetzt wird. Und die Zeit selbst ist keine Regelgröße, wie Geschwindigkeit oder Querablage. Der Geschwindigkeitsfehler bleibt wie alle anderen Abweichungen in den Zustandsgrößen über eine Neuplanung hinaus bestehen.

## 7.3 Statische Objekte

Bei Ausweichtrajektorien um Objekte herum stellt sich die Frage, auf welcher Seite das Objekt überholt werden soll. Gradientenbasierte Optimierungsverfahren laufen von der Startlösung in ein lokales Optimum. Daher lässt sich oft schon anhand der Startlösung bestimmen, auf welcher Seite die Lösung liegen wird, und es bringt keinen Vorteil, die Entscheidung dem Optimierer zu überlassen. Wenn eine vorgeschaltete Taktik-Planung oder Verkehrsregeln eine Seite bestimmen, wird die Entscheidung übernommen. Anderenfalls entscheidet eine vorgeschaltete Logik, welche Seite ausreichend Platz bietet und zu einer geringeren Abweichung von der vorherigen Planung bzw. der Ideallinie führt. Bei dynamischen Objekten ist diese Entscheidung von der zu optimierenden Ego-Dynamik abhängig. Dann muss zusätzlich die Zeit für das Ego-Fzg und die Objekte parametrisiert über  $s$  berücksichtigt werden, was nicht Bestandteil der vorliegenden Arbeit ist.

Da der Querversatz  $d$  eine Optimierungsvariable ist und das Problem über den Weg parametrisiert ist, stellt ein statisches Objekt eine Box-Restriktion dar, die keinen zusätzlichen Aufwand bringt, was ein Vorteil dieser Methode ist. Ein Potential in Form eines lokal definierten kubischen Polynoms in  $d$  drückt das Fzg zusätzlich vom Objekt weg, um im Kompromiss zur Fahrzeit den Sicherheitsabstand zu erhöhen. Diese Objektbeschreibung verändert die Struktur des Optimierungsproblems nicht, sodass beliebig viele Objekte mit einbezogen werden können.



**Bild 7.2:** Die Weglänge des Planungshorizonts  $S_{Ab}$  variiert entlang der Strecke in Abhängigkeit der Diskretisierung. Seine Dauer  $T_{Ab}$  hängt vom geplanten Geschwindigkeitsverlauf  $v$  ab. Der Plot zeigt beispielhaft Werte für die Ideallinie auf dem Autódromo do Algarve.

## 7.4 Ergebnisse der Echtzeit-Planung

Dieses Kapitel präsentiert die Ergebnisse der zyklischen Echtzeit-Planung für den Golf GTI auf dem 4,5 km langen *Autódromo do Algarve*. In den Abschnitten 7.4.1 und 7.4.2 werden Simulationsergebnisse aus ADTF dargestellt, anschließend in Abschnitt 7.4.3 Messergebnisse einer realen Erprobungsfahrt. Während der Fokus bei den Simulationsergebnissen auf der echtzeitfähigen Planung von Ausweichtrajektorien bei plötzlich erscheinenden Hindernissen liegt, stehen bei den Messergebnissen fahrdynamische Größen im Vordergrund, um die Güte der Planung und des Fzg-Modells zu beurteilen.

Die Diskretisierung der Referenz beträgt wiederum 1...5 m, woraus insgesamt  $N_{Rk} = 1510$  DP resultieren. Der Planungshorizont der zyklischen Planung beträgt konstant  $N_{Ab} = 95$  DP. Bild 7.2 zeigt exemplarisch die räumliche und zeitliche Länge des Planungshorizonts entlang der Strecke. Sie beträgt für diese Trajektorie überall mehr als  $S_{Ab} = 126$  m bzw.  $T_{Ab} = 6,58$  s. Aus der Trajektorie resultiert eine zeitliche Diskretisierung, die im Mittel 90 ms und maximal 170 ms beträgt.

Der Planungshorizont ist im Vergleich zu Ansätzen in der Literatur relativ lang, bspw. verwendet der Querführungsansatz mittels QP [34] einen Horizont von 4 s bei einer Diskretisierung von 200 ms und die Brems-Ausweich-Assistenz mittels nichtlinearer MPC [90] einen 2 s-Horizont. Je länger der Planungshorizont ist, umso geringer wirkt sich die Wahl des Endzustands  $\mathbf{x}(S_{Ab})$  bzw. des MAYER-Terms  $\vartheta(\mathbf{x}(S_{Ab}))$  auf den relevanten Teil, also den Beginn der Trajektorie aus. Da man in der Realität damit rechnen muss, dass diese Wahl suboptimal sein kann (vgl. Abschnitt 7.2), soll ihr Einfluss minimiert werden. Auf der anderen Seite darf die Rechenzeit nicht zu stark ansteigen (vgl. Abschnitt 7.1). Für die Rennstrecke hat sich in unserem Projekt die verwendete Horizontlänge als guter Kompromiss erwiesen.



**Tabelle 7.1:** Statische Objekte entlang der Strecke mit folgenden Angaben: $s_{\text{obj}}$ : Position des Objekt-Mittelpunkts,  $l_{\text{obj}}$ : Länge des Objekts,  $\Delta s_{\text{view}}$ : Sichtbarkeit, $\Delta d_{\text{ideal}}$ : Mindestquerablage zur Ideallinie für Kollisionsfreiheit, $\Delta t_{\text{view}}$ : Zeit bis zur Kollision (TTC), $\Delta t_{\text{opti}}$ : TTC abzüglich der erlaubten Rechenzeit unter Berücksichtigung der Diskretisierung, $v_{\text{view}}$ : Fzg-Geschwindigkeit beim Erkennen des Objekts, $\bar{z}_B = -\frac{\bar{a}_x}{g} = -\frac{v_{\text{view}}}{g \Delta t_{\text{view}}}$ : Mittlere Abbremsung für eine theoretische Notbremsung.

Nr.	$\frac{s_{\text{obj}}}{\text{m}}$	$\frac{l_{\text{obj}}}{\text{m}}$	$\frac{\Delta s_{\text{view}}}{\text{m}}$	$\frac{\Delta d_{\text{ideal}}}{\text{m}}$	$\frac{\Delta t_{\text{view}}}{\text{s}}$	$\frac{\Delta t_{\text{opti}}}{\text{s}}$	$\frac{v_{\text{view}}}{\text{km/h}}$	$\bar{z}_B$
1	345	6	110	-6,7	2,33	2,13	189	2,3
2	708	5	70	5,7	2,43	2,38	128	1,5
3	1255	10	80	5,8	1,89	1,77	152	2,3
4	1368	4	40	-6,4	1,85	1,73	80	1,2
5	1739	3	65	-1,5	1,70	1,55	140	2,3
6	2071	6	65	-3,9	2,16	2,15	92	1,2
7	2105	9	80	4,6	3,17	3,06	100	0,9
8	2676	6	50	-4,0	2,18	2,06	74	1,0
9	3152	5	45	1,8	1,78	1,64	83	1,3
10	4000	8	80	-5,7	1,87	1,86	147	2,2
11	4124	8	50	-	-	-	152	-

vorgegebene oder vorab bekannte Daten

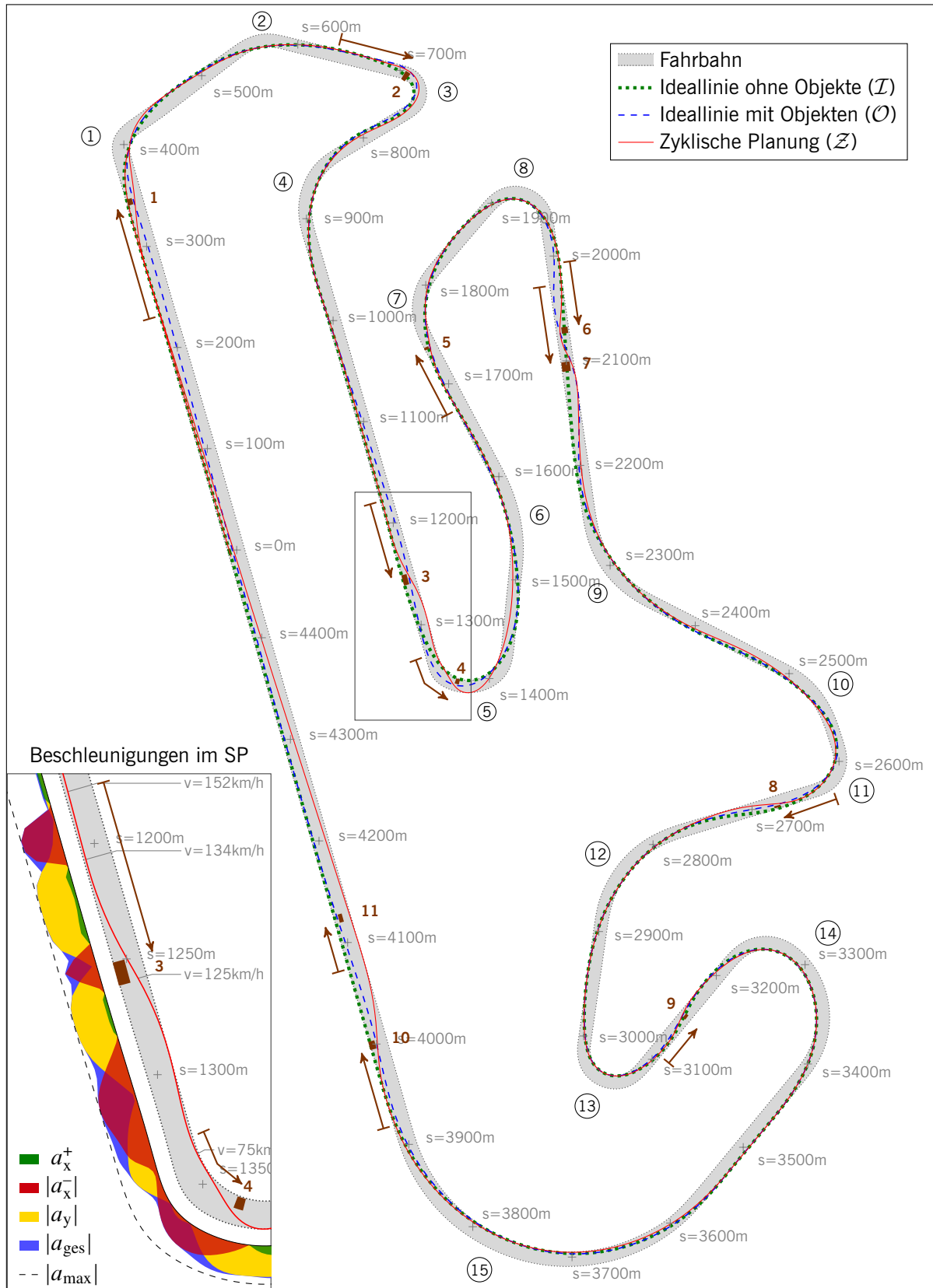
simulierte Daten

### 7.4.1 Simulationsergebnisse der zeitoptimalen Planung

Entlang der Strecke sind 11 Objekte gemäß Tabelle 7.1 positioniert, die in Bild 7.3 als braune Rechtecke abgebildet sind. Die Pfeile in Bild 7.3 visualisieren den Bereich  $\Delta s_{\text{view}}$ , in welchem die Objekte für die Planung sichtbar sind. Mit Ausnahme von Objekt 11 sind sie so positioniert, dass das Fzg mindestens  $\Delta d_{\text{ideal}}$  von der Ideallinie abweichen muss, um eine Kollision zu vermeiden. Die Zeit vom ersten Erkennen bis zum Erreichen des Objekts, wenn das Fzg auf der zuletzt geplanten Trajektorie weiterfährt, wird mit  $\Delta t_{\text{view}}$  oder auch „time to collision“ (TTC) bezeichnet.<sup>1</sup>  $\Delta t_{\text{opti}}$  gibt an, wie viel Zeit der Trajektorienplanung tatsächlich für das Ausweichmanöver bleibt: Die Zeit beginnt beim ersten *veränderbaren* DP, also dem DP nach der maximal zulässigen Rechenzeit (vgl. Abschnitt 7.1), und endet beim letzten DP vor dem Objekt. Wichtig ist die Annahme in dieser Simulation, dass die Steuergeräte keine Totzeit aufweisen. Alle Objekte erscheinen so kurzfristig, dass eine Notbremsung unmöglich ist, weil die notwendige mittlere Abbremsung  $\bar{z}_B$  zu hoch wäre. Lediglich bei Objekt 7 und 8 läge die Verzögerung bei max. 1g, jedoch sind dort bereits hohe Querbeschleunigungen, die das Notbremsmanöver unmöglich machen.

Bild 7.3 zeigt 3 Trajektorien: grün gepunktet die Ideallinie ohne Objekte ( $\mathcal{I}$ ), blau gestrichelt die offline berechnete Ideallinie mit Objekten ( $\mathcal{O}$ ) für  $\Delta s_{\text{view}} \rightarrow \infty$  und rot durchgezogen die zykli-

<sup>1</sup>Im Gegensatz zur klassischen Definition der TTC, bei der die Geschwindigkeit als konstant angenommen wird, also  $\text{TTC} = \Delta s_{\text{view}} / v_{\text{view}}$ , siehe [92, Kap. 47.6.1], wird die TTC hier über die zuletzt geplante Trajektorie ermittelt.



**Bild 7.3:** Ideallinie für das Autódromo do Algarve mit und ohne Objekte sowie die resultierende Linie einer zyklischen Planung. Die Fahrbahn­ränder sind abzüglich der halben Fzg-Breite gezeichnet. Die Pfeile zeigen den Sichtbarkeitsbereich der Objekte für die zyklische Planung.

sche Echtzeit-Planung ( $\mathcal{Z}$ ). Das Höhenprofil der Rennstrecke wurde darin nicht berücksichtigt. In diesem Kapitel wird auf eine Analyse der beiden offline berechneten Trajektorien verzichtet, weil die Abschnitte 5.5 und 6.2.4 bereits entsprechende Analysen enthalten. Zudem behandelt [95] dieselbe Strecke mit teils ähnlich positionierten Hindernissen und fast identischen Fzg-Parametern.

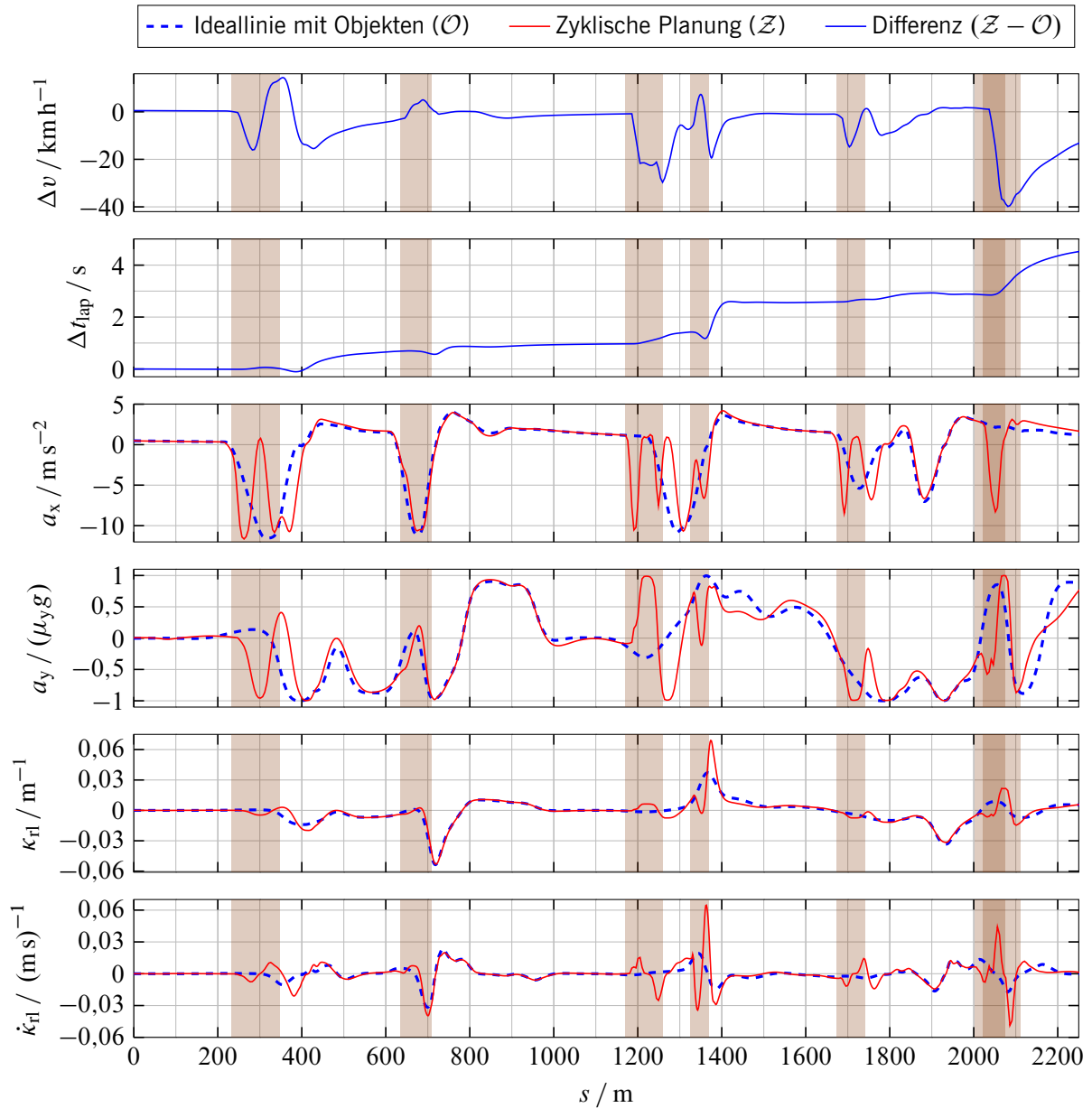
Bei der zyklischen Planung wird das Fzg simuliert, als ob es dem Sollverlauf exakt folgt. Nach Ablauf eines Planungsschritts wird nach einer kurzen Pause, die hier 40 ms beträgt, die nächste Planung an der simulierten Fzg-Position aufgerufen. Nach einer vollständigen Runde beginnt die letzte Planung also bei  $s = 0$  m und endet 95 DP später bei  $s = 400$  m (vgl.  $S_{\text{Ab}}$  in Bild 7.2). Zur anschließenden Datenanalyse werden die entsprechenden Abschnitte aller Trajektorien zu der tatsächlichen Trajektorie zusammengesetzt. Die Berechnung erfolgt auf einer INTEL Core i7 Mobile-CPU. Im Mittel lag eine Neuplanung nach 1,1 DP vor, in 99,7 % nach spätestens 3 DP. Das Fzg muss sich mindestens 1 DP fortbewegt haben, bevor eine Neuplanung durchgeführt wird.

Aufgrund der endlichen und sogar recht kurzen Sichtbarkeit der Objekte, ergibt  $\mathcal{Z}$  eine schlechtere Linie als  $\mathcal{O}$ . Zudem bestraft der MAYER-Term  $\vartheta(\mathbf{x}(S_{\text{Ab}}))$  im Optimierungsproblem (7.1b) nur die Abweichung zu  $\mathcal{I}$ , denn  $\mathcal{O}$  ist ja nicht bekannt. Somit liefert die Optimierung zwar das mathematische Optimum, aber im Sinne der Rundenzeit eine suboptimale Lösung: Ohne Objekte dauert eine Runde  $T_{\text{lap}, \mathcal{I}} = 136,2$  s, mit Objekten  $\Delta T_{\text{lap}, \mathcal{I} \rightarrow \mathcal{O}} = 1,7$  s länger. Die zyklische Planung benötigt noch  $\Delta T_{\text{lap}, \mathcal{O} \rightarrow \mathcal{Z}} = 6,1$  s mehr.

Die Bilder 7.4 und 7.5 zeigen jeweils einige relevante Größen von  $\mathcal{O}$  und  $\mathcal{Z}$ , wobei die Strecke zur besseren Übersicht auf zwei Bilder aufgeteilt wurde. Dort wo die Objekte keinen Einfluss haben, weist  $\mathcal{Z}$  einen sehr ähnlichen Verlauf zu  $\mathcal{O}$  mit fast identischer Fahrzeit auf, was zeigt, dass das Optimierungsproblem mit der gewählten Endpunktbewertung sehr gut geeignet ist.

Bei der Geschwindigkeitsdifferenz fällt auf, dass es Bereiche gibt, in denen  $\Delta v > 0 \Leftrightarrow v_{\mathcal{Z}} > v_{\mathcal{O}}$  ist, was einer Erklärung bedarf: Bspw. in Kurve 15 (*Galp Corner*) bei  $s \geq 3600$  m, wo kein Objekt die Trajektorie beeinflusst. Optimal ist es, die Kurve spät und somit *etwas langsamer* als eine gewöhnliche Kurve anzufahren, um am Kurvenausgang mit höherer Geschwindigkeit auf die Gerade zu fahren.  $\mathcal{Z}$  plant aber nur  $S_{\text{Ab}} \approx 300$  m in die Zukunft (Bild 7.2), sodass die Information über die lange Gerade nur über den MAYER-Term einfließt. Aus Sicherheitsgründen wird am Endpunkt nie gefordert, dass die Geschwindigkeit der Offline-Lösung erreicht wird, sondern nur dass sie nicht überschritten wird. Daher ist es für  $\mathcal{Z}$  optimal, den ersten Teil der Kurve schneller zu fahren, auch wenn der Kurvenausgang dadurch ein klein wenig langsamer ist. Dass sich das wegen der Länge der Geraden negativ auf die Gesamtzeit auswirkt, bildet das Optimierungsproblem der zyklischen Planung mathematisch nicht ab. Bild 7.5 zeigt, dass die Geschwindigkeit bei  $s \geq 3800$  m auch nur minimal niedriger ist, und dennoch wäre die Bilanz am Ende der Geraden etwas schlechter, was in der Simulation aufgrund der Objekte 10 und 11 nicht mehr nachzuvollziehen ist.

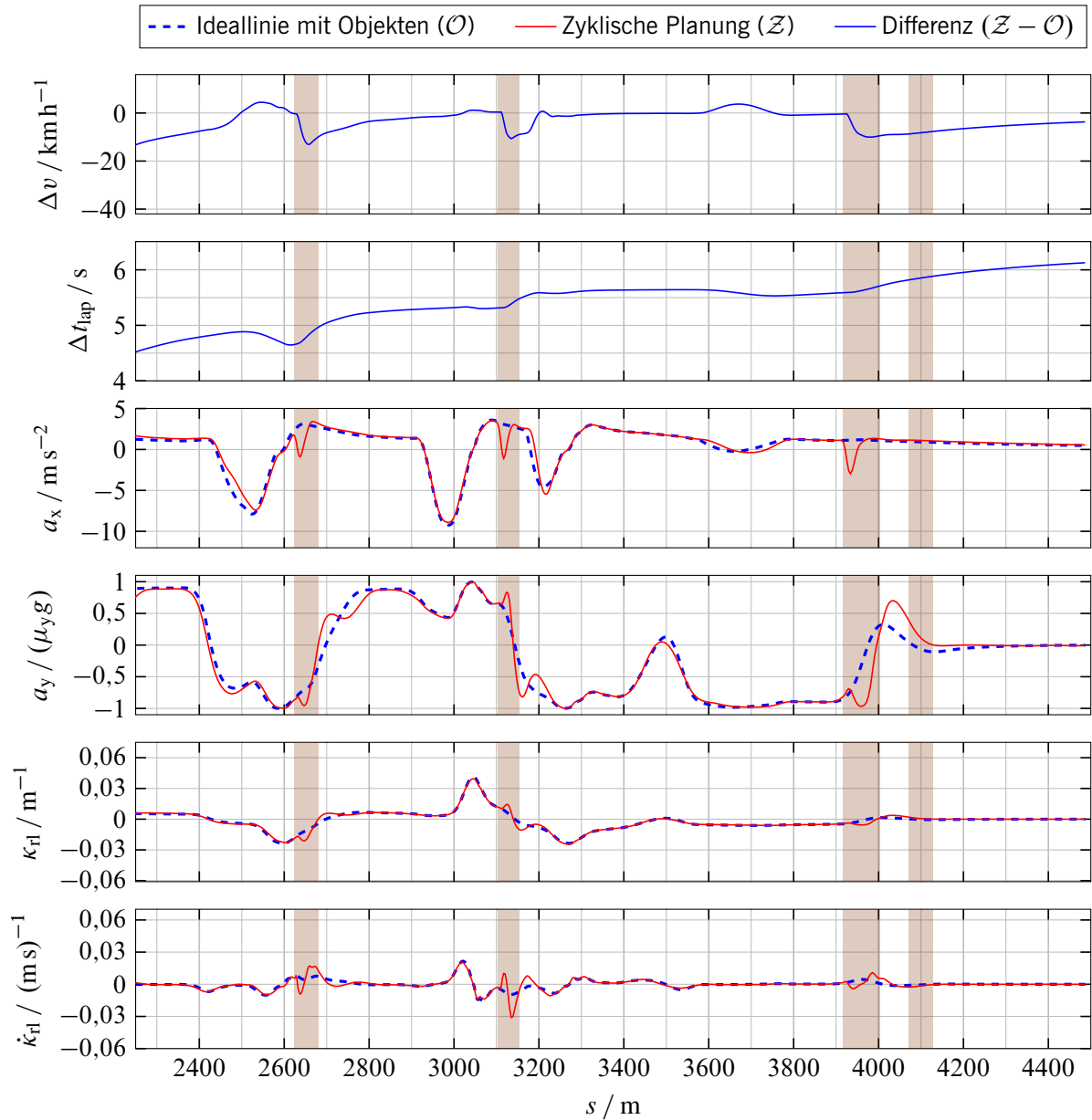
Eine andere Situation, bei der  $v_{\mathcal{Z}} > v_{\mathcal{O}}$  ist, ergibt sich bei Objekt 1. Dort muss beim Erkennen des Objekts stark verzögert werden, damit das Ausweichmanöver überhaupt durchführbar ist. Es erfordert in Höhe des Objekts einen weitgehend geraden Abschnitt (vgl. Bild 7.3) und verzögert den Bremspunkt, wodurch sich dort  $\Delta v > 0$  ergibt. Der späte Bremspunkt führt jedoch zu ei-



**Bild 7.4:** Geschwindigkeits- und Fahrzeitdifferenz zwischen der Ideallinie mit Objekten und der zyklischen Planung sowie Beschleunigung und Krümmung beider Trajektorien für  $s \leq 0 \dots 2250$  m. In den braun hinterlegten Bereichen sind die jeweiligen Objekte 1 – 7 für die Planung sichtbar.

nem engen Radius in Kurve 1, was eine niedrigere Geschwindigkeit  $\Delta v < 0$  und insgesamt einen Zeitverlust  $\Delta t_{\text{lap}} \approx 0,7$  s mit sich bringt.

Generell ergibt sich ein sehr glatter Krümmungsverlauf  $\kappa_{rl}$ , obwohl fast jeder DP der Trajektorie aus einer einzelnen Planung stammt. Selbst die numerisch berechnete Zeitableitung  $\dot{\kappa}_{rl}$  ist gewohnt glatt. Peaks treten nur im Bereich einiger Objekte auf, was darauf zurückzuführen ist, dass die TTC so klein ist, dass hohe Beschleunigungsänderungen erforderlich sind, um eine kollisionsfreie Trajektorie zu finden. Dies zeigt die Vergrößerung in Bild 7.3 im Bereich der Objekte 3 und 4:



**Bild 7.5:** Fortsetzung von Bild 7.4 für  $s \geq 0 \dots 2250 \text{ m}$ . In den braun hinterlegten Bereichen sind die jeweiligen Objekte 8 – 11 für die Planung sichtbar.

Entlang der Fahrbahn sind folgende SP-Beschleunigungen abgebildet: positive Längsbeschleunigung  $a_x^+$  (grün), Betrag der Verzögerung  $|a_x^-|$  (rot), Betrag der Querschleunigung  $|a_y|$  (gelb) sowie die resultierende Gesamtbeschleunigung  $|a_{\text{ges}}|$  (blau). 5 m (120 ms) nach Erkennen von Objekt 3 wird maximal verzögert, um ca. 20 km/h abzubauen. Gleichzeitig wird schon das Ausweichmanöver eingeleitet. Die kurze Phase um  $s = 1250 \text{ m}$ , in der keine Querschleunigung benötigt wird, wird wieder zum Bremsen genutzt. Längs- und Querschleunigungsphasen wechseln sich mit Überblendungen so optimal ab, dass das Kraftpotential voll ausgenutzt wird. Dass  $a_{\text{ges}}$  manchmal nicht ganz am Limit  $a_{\text{max}}$  ist, hat zwei Gründe: Zum einen erlaubt die Aktorik keine noch schnelleren Übergänge und zum anderen kann eine der beiden Achsen schon an ihrem Limit sein, bevor die (theoretisch) maximale SP-Beschleunigung erreicht ist ( $a_{\text{max}}$  wurde hier un-

ter der Annahme berechnet, dass beide Achsen ihr Limit gleichzeitig erreichen). Auffällig ist die positive Längsbeschleunigung  $a_x^+ > 0$  während der Lenkphasen: Die Beschleunigung in Richtung der Fzg-Längsachse dient dazu, die Geschwindigkeit  $v$  (in Richtung des Kurswinkels) trotz der hohen Querbeschleunigung (großer Schwimmwinkel) weitgehend konstant zu halten, also  $\dot{v} \approx 0$  in (3.34a) auf S. 36. Solange die Geschwindigkeit fahrbar ist, ist das sinnvoll, weil sich  $v = \text{const}$  gemäß (3.31) positiv auf einen gleichmäßigen Krümmungsverlauf auswirkt. So berechnet die Planung also von selbst ohne spezielle Vorgaben ein perfektes Brems-/Ausweichmanöver.

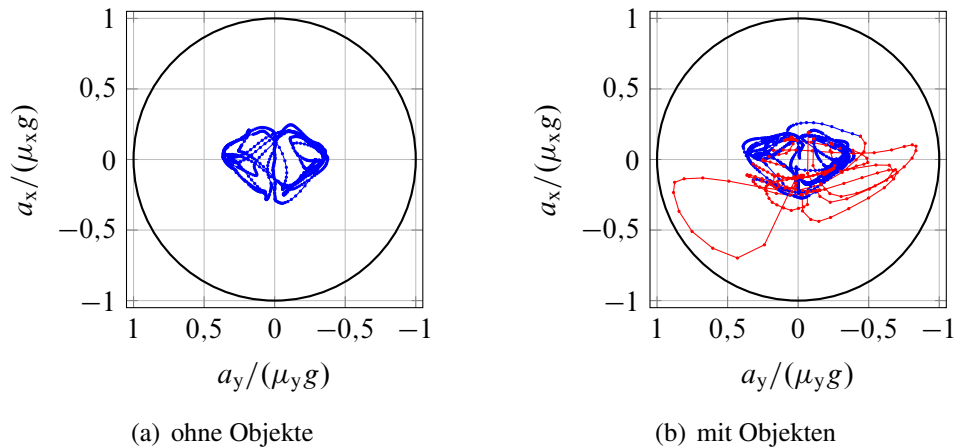
Diese Simulation soll zeigen, wie sich die Planung in extrem kritischen Situationen verhält, in denen es nur einen kleinen Bereich im Zustandsraum gibt, in dem die Lösung liegen muss. Verschärft man die Situation noch etwas, existiert gar keine Trajektorie mehr, die alle Nebenbedingungen erfüllt. Allerdings wäre es schlicht unrealistisch, zu glauben, dass so eine Trajektorie im realen Fzg immer so abgefahren werden kann: Zum Bsp. Modellungenauigkeiten bei der Planung, Schwingungen im Regelkreis durch nicht modellierte Nichtlinearitäten, Achslastschwankungen durch Anregungen im Fahrwerk und der Fahrbahn oder Verzögerungen in der Ausführung der Software im Bereich einiger Hundertstel Sekunden können dazu führen, dass das Manöver scheitert und das Fzg ausbricht oder mit dem Hindernis kollidiert. Trotzdem ist es wichtig, dass die Planung immer zuverlässig bis ans theoretische Limit gehen kann, denn im gesamten System darf die Planung als erstes Glied in der Kette niemals die Systemgrenze bilden (vgl. Einleitung in Kapitel 1).

Objekt 11 ist so positioniert, dass es kein Ausweichmanöver veranlasst, aber auf beiden Seiten überholt werden kann. Die Logik für  $\mathcal{O}$  entscheidet sich für die linke Seite, weil dies zu der geringeren Abweichung zu  $\mathcal{I}$  führt. Die Logik für  $\mathcal{Z}$  bewertet hingegen nicht die Abweichung zu  $\mathcal{I}$ , sondern zu der Trajektorie, die beim Erkennen des Objekts vorliegt. Sie ist aufgrund von Objekt 10 bereits am rechten Fahrbahnrand, wodurch Objekt 11 rechts überholt wird. Im weiteren Verlauf tendiert  $\mathcal{Z}$  langsam wieder zum linken Rand, was auf die Endpunktbewertung zurückzuführen ist.

Bei der Auswertung der Rechenzeit ist zu berücksichtigen, dass die Optimierung einer harten Echtzeitbedingung unterliegt (Abschnitt 7.1), sodass sie abgebrochen wird, wenn die prädizierte Maximal-Rechenzeit überschritten wird. Dies geschah in dieser Simulation in 0,8 % der Zyklen. In jedem dieser Fälle war die darauffolgende Planung erfolgreich. Für die folgende statistische Auswertung wurden dann beide Rechenzeiten summiert, was zu dem hohen Maximalwert führt. Es ergibt sich:

Mittelwert:	39 ms,	Minimum:	17 ms,	Maximum:	336 ms,
0,9-Quantil:	60 ms,	0,95-Quantil:	85 ms,	0,99-Quantil:	133 ms

Im Vergleich mit der eingangs angeführten MPC zur Fzg-Querführung [34], die eine Zykluszeit von 20 ms auf einer AutoBox erzielt, sind diese Zykluszeiten höher, jedoch beträgt der Planungshorizont in [34] nur 20 DP (statt hier 95 DP). Die Brems-Ausweich-Assistenz mittels nichtlinearer MPC [90] läuft auf einer INTEL Core i5 Mobile-CPU in einem 100 ms-Zyklus, wobei die Rechenzeit für den 2 s-Horizont die tatsächliche Fahrzeit in einer Simulation um bis zu 20 % überschreitet.



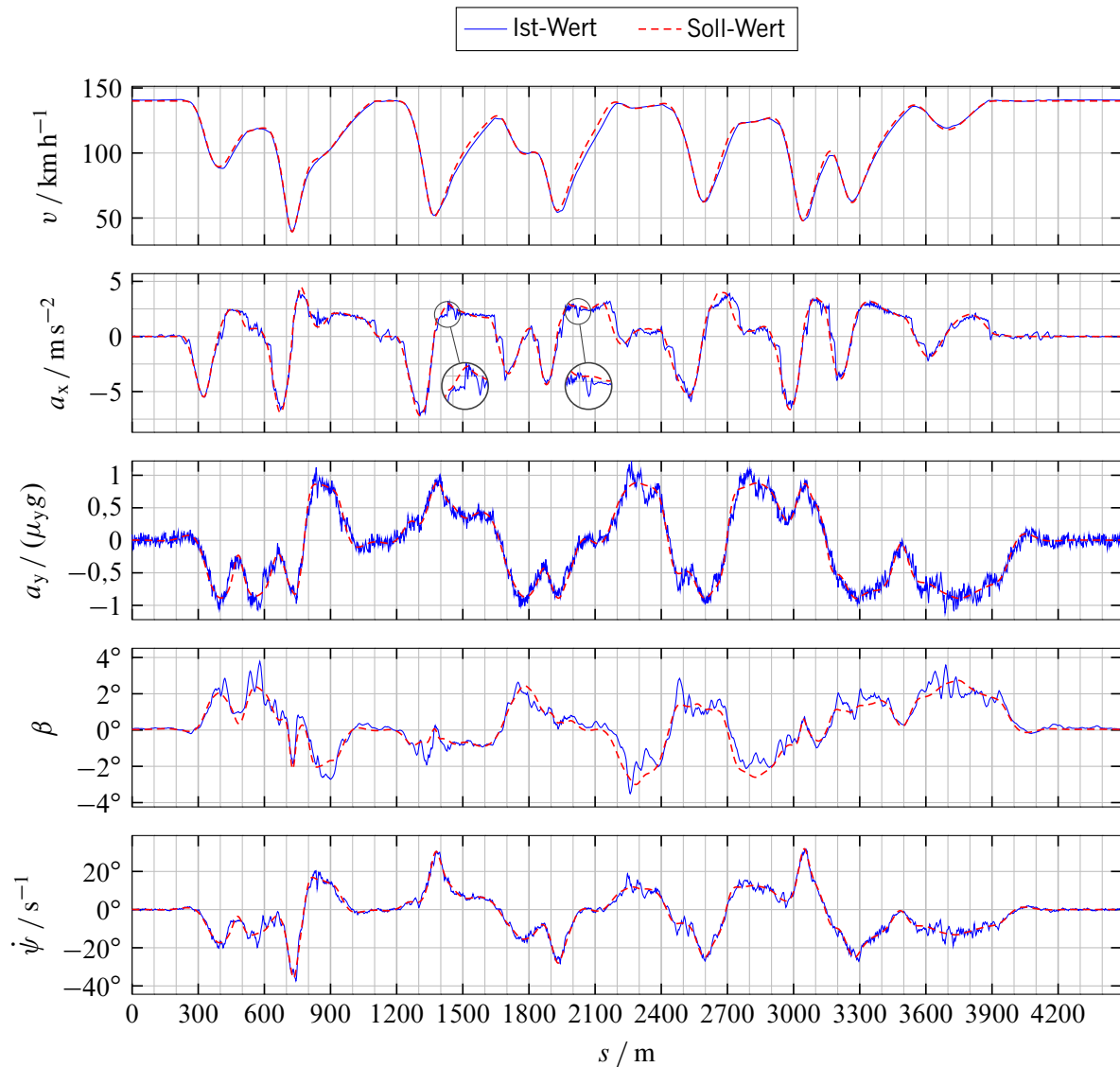
**Bild 7.6:** Das GG-Diagramm der Komfort-Planung geht bei Ausweichmanövern bis an den Kammschen Kreis.

### 7.4.2 Simulationsergebnisse zu Ausweichmanövern in der Komfort-Planung

Kritische Ausweichmanöver, die spontan bis ans Limit geplant werden müssen, sind nicht nur in der zeitoptimalen Planung möglich, sondern auch in der „Komfort-Planung“, die in Abschnitt 5.6 auf S. 94 ff. vorgestellt wurde. Dass die Komfort-Planung in der Lage ist, bis ans Limit zu gehen, lässt sich nur in einer Online-Planung mit plötzlich erscheinenden Hindernissen zeigen. Sie wurde mit derselben Parametrierung wie in Bild 5.13(d) durchgeführt, was ohne Objekte zu dem GG-Diagramm Bild 7.6(a) führt. Für die Ausweichmanöver wurde die Sichtbarkeit der Objekte  $\Delta s_{\text{view}}$  aus Tabelle 7.1 so reduziert, dass sich trotz des niedrigeren Geschwindigkeitsniveaus kritische Werte für die TTC ergeben. An Bild 7.6(b) erkennt man, wie die Planung in den Abschnitten ohne Objekte (blau) im komfortablen Bereich bleibt, und in den Abschnitten, wo Objekte sichtbar sind (rot), den Komfortbereich verlässt. Am Ende eines Ausweichmanövers verläuft die Trajektorie wieder in den gewünschten Bereich zurück. Die Rechenzeiten der Komfort-Planung sind etwas kürzer als bei der zeitoptimalen Planung, wenn sich die Trajektorie im Komfortbereich befindet und nicht an einer UNB des Optimierungsproblems.

### 7.4.3 Messergebnisse einer Realfahrt auf dem Autódromo do Algarve

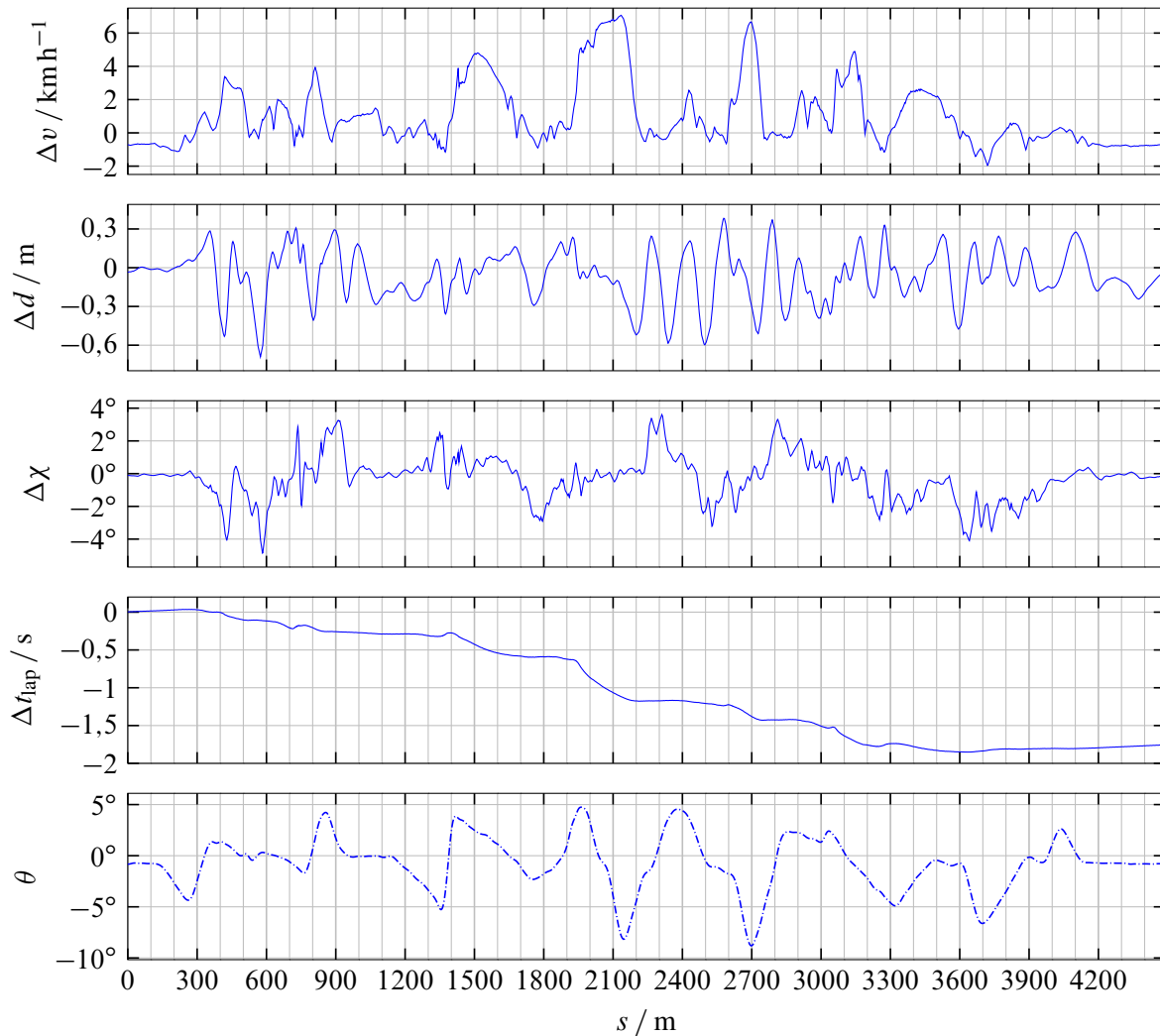
Im Gegensatz zur Simulation in Abschnitt 7.4.1 wurde bei der Erprobungsfahrt ein Sicherheitsabstand von 0,5 m zum Fahrbahnrand eingeplant, die Geschwindigkeit auf  $v_{\text{max}} = 140 \text{ km/h}$  begrenzt und das Höhenprofil berücksichtigt. Der Kraftschlussbeiwert wurde auf  $\mu = 0,9$  geschätzt. Die simulierte Rundenzeit liegt daher nur noch bei  $T_{\text{lap}} = 150,2 \text{ s}$ , also 14 s langsamer als in Abschnitt 7.4.1. Bild 7.7 stellt relevante Größen der Soll-Trajektorie den Messgrößen gegenüber. Ist die Limitierung der Geschwindigkeit aktiv, beeinflusst sie die (Soll-)Trajektorie in der Umgebung stark, was sich an einem untypisch zaghaften und späten Anbremsen sowohl am Ende der Geraden bei Kurve 1 (*Primeira*) als auch bei Kurve 5 (*VIP Tower*, vor  $s \approx 1400 \text{ m}$ ) zeigt. Auch die Querschleunigung in der letzten Kurve 15 (*Galp*) fällt daher niedriger aus als in Bild 7.5.



**Bild 7.7:** Zusätzlich zur Position und Orientierung (nicht abgebildet) überträgt die Planung die Geschwindigkeit und Beschleunigung an die Regelung. Schwimmwinkel und Gierrate bilden keine Regelgrößen. Die gemessenen Werte stammen z. T. aus einer Fusion von GPS und Intertialsensorik sowie aus einem regelungstechnischen Beobachter.

Im Allgemeinen kann der Regler der gewünschten Beschleunigung und auch ihrem Gradienten (Ruck) ohne nennenswerte Abweichungen folgen. Der Geschwindigkeitsverlauf weist allerdings während längerer Beschleunigungsphasen einen Schleppfehler auf, wie man auch in Bild 7.8 an  $\Delta v$  erkennen kann. Um der näherungsweise rampenförmigen Geschwindigkeitsvorgabe ohne Schleppfehler folgen zu können, wäre ein doppelter I-Anteil im Regler notwendig, auf welchen hier verzichtet wurde, um zum einen die Dynamik des Regelkreises nicht zu beeinträchtigen und zum anderen einen Wind-up-Effekt von vornherein auszuschließen, welcher im Grenzbereich besonders relevant ist. Meist entsteht ein größerer Geschwindigkeitsfehler dann, wenn zu Beginn der Beschleunigungsphase eine Störung auf  $a_x$  wirkt: Dies wird v. a. durch die Steigung (s.  $\theta$  in Bild 7.8) oder durch Schaltvorgänge hervorgerufen, welche  $a_x$  trotz Doppelkupplungsgetriebe





**Bild 7.8:** Die Geschwindigkeitsabweichung  $\Delta v$ , die Querablage  $\Delta d$  und die Kurswinkelabweichung  $\Delta \chi$  gehören zur Regeldifferenz. Darunter ist die Differenz zwischen der geplanten und der realen Fahrzeit abgebildet. Das unterste Diagramm zeigt den (interpolierten) Fahrbahnneigungswinkel  $\theta$  entlang der Fahrlinie.

kurzfristig reduzieren (s. Vergrößerungen in Bild 7.7). Auch wenn der Geschwindigkeitsfehler im Mittel lediglich  $|\overline{\Delta v}| = 1,5 \text{ km/h}$  beträgt, kumulieren sich die Abweichungen in der Rundenzeit zu  $\Delta T_{\text{lap}} = 1,8 \text{ s}$ , was  $1,2 \%$  entspricht. Für einen Rennfahrer wäre das viel, für unser Projektziel, ein System zu entwickeln, das automatisiert und zuverlässig im Grenzbereich fahren kann, ist es weniger relevant. Zum Vergleich war die mit einem Zweispurmodell berechnete Trajektorie aus [20, S. 183] ca.  $5 \%$  schneller als die Bestzeit professioneller Rennfahrer.

Eine Querablage  $\Delta d$  und ein Kurswinkelfehler  $\Delta \chi$  treten lokal immer gemeinsam auf, weil eine Korrektur in einer der beiden Größen einer Abweichung in der anderen bedarf. Die Querablage  $\Delta d$  beträgt im Mittel  $|\overline{\Delta d}| = 16 \text{ cm}$  und maximal  $|\Delta d|_{\text{max}} = 70 \text{ cm}$ . Der Kurswinkel weicht im Mittel  $|\overline{\Delta \chi}| = 1^\circ$  und maximal  $|\Delta \chi|_{\text{max}} = 4,9^\circ$  ab. Dabei ist zu berücksichtigen, dass bei der Regelung des Kurswinkels der Ist-Schwimmwinkel  $\beta$  nicht verwendet wird, da er bei einem spä-

teren Serieneinsatz auch nicht in ausreichend verlässlicher Qualität zur Verfügung stehen wird. In Anbetracht dass sich das Fzg dabei an seinem fahrdynamischen Limit bewegt, kann es der Fahrlinie also gut folgen.

Für die Beurteilung der Fahrdynamik und der Güte des Fzg-Modells sind Analysen von Schwimmwinkel und Gierrate sehr interessant. Denn zum einen ist gerade der Einfluss der Reifenparameter auf diese Verläufe extrem groß, während die Linie selbst weniger beeinflusst wird. Zum anderen geben sie Aufschluss darüber, wie gut die Trajektorie fahrbar ist, weil im Grenzbereich jede (Regler-)Korrektur einer lateralen Abweichung zu einem starken Anstieg von Schlupf und Schwimmwinkel führt, da die Reifen kaum noch Potential haben und schnell anfangen zu rutschen. Man sieht z. B., dass in Kurve 2 ( $s \approx 550$  m), wo eine hohe Querablage vorliegt, auch relativ hohe Schwimmwinkel entstehen. Allgemein passen die in der Optimierung berechneten Verläufe von  $\beta$  und  $\dot{\psi}$  aber sehr gut zu den gemessenen. Man kann erkennen, dass sich das Fzg aufgrund der hohen Beschleunigungen und Schwimmwinkel nahe am Limit bewegt, während es der Ideallinie mit relativ kleiner Querablage folgt und dabei das Limit nie überschreitet. Somit wurde das Projektziel, eine Trajektorie zuverlässig bis zum fahrdynamischen Limit planen und automatisiert abfahren zu können, in vollem Umfang erreicht.

## 8 Zusammenfassung und Ausblick

Eine zeitoptimale Trajektorienplanung für automatisiertes Fahren bis in den fahrdynamischen Grenzbereich wurde entworfen, in MATLAB sowie echtzeitfähig in C++ implementiert und sowohl in Simulationen als auch bei Erprobungsfahrten erfolgreich getestet. Zu Beginn der vorliegenden Arbeit wird das zugrundeliegende Konzept der Planung im Kontext diverser Ansätze aus der Literatur erläutert. Die wesentlichen Punkte darin sind, dass

- der Lösungsraum der Planung nicht a priori auf vorgegebene Trajektorien oder Bewegungsmuster eingeschränkt ist,
- Quer- und Längsplanung simultan, also in einer gemeinsamen Optimierung, erfolgen und
- alle Zustände und Zustandsgleichungen über den Weg parametrisiert sind.

Die wegabhängige Parametrierung erfordert eine nichtlineare Transformation aller zeitabhängigen Differentialgleichungen, welche anhand kinematischer Beziehungen erfolgt. Sie ermöglicht es, ein Gütefunktional aufzustellen, das die tatsächliche Fahrzeit entlang der Wegstrecke beschreibt, ohne ein Optimierungsproblem mit freier Endzeit und freiem Endzustand zu lösen. Zudem hat sie den Vorteil, dass statische Elemente, wie Verlauf und Breite der Fahrbahn, im mathematischen Problem statisch bleiben. Dabei wird die Trajektorie ausschließlich in lokalen Koordinaten angegeben, welche sich auf eine global definierte Referenzlinie entlang der Strecke beziehen.

Als Fahrzeugmodell dient ein nichtlineares Einspurmodell mit einer PACEJKA-Reifenkennlinie, welches um eine quasistatische Achslastverteilung und eine dynamische Bremskraftverteilung erweitert wird. Bei der Modellbildung ist es wichtig, einen möglichst guten Kompromiss zwischen Genauigkeit und späterer Rechenzeit bei der Optimierung zu finden. Hierfür werden in gewissen Teilmodellen Approximationen durchgeführt, die nur geringen Einfluss auf die Genauigkeit haben, aber die Komplexität aus Sicht der Optimierung signifikant reduzieren. Ein wesentlicher Aspekt hierbei ist die Modellierung der Achslast, die in Abschnitt 3.2.5 hinsichtlich Modellgüte und Rechenzeit simulativ untersucht wurde.

Auch zur Modellierung der Aktordynamik (Motor, Bremse, Lenkung) dient ein spezieller Ansatz, der ausgehend von der Annahme eines  $PT_1$ - bzw.  $PT_2$ -ähnlichen Verhaltens die Dynamik so modelliert, dass sie besonders effizient ins Optimierungsproblem eingebaut wird. Zusätzlich wird sie nicht als Nebenbedingung, sondern als Gütefunktion formuliert, welche zu einer geschwindigkeitsabhängigen quadratischen Form führt. Sie erhöht nicht nur die Konvergenz, sondern führt auch zu sehr glatten Trajektorien, deren Krümmung bis mindestens zur zweiten Ableitung stetig ist (Abschnitt 5.1.2), was im Fahrverhalten deutlich spürbar ist. In Erprobungsfahrten hat sich gezeigt, dass der Trajektorienfolgeregler dem so berechneten Führungsgrößenverlauf sehr gut folgen kann. Messfahrten zeigen zudem, dass auch bei internen Größen des Modells, wie Schwimmwinkel und Gierrate, eine sehr gute Korrelation zwischen den Werten in der Planung und den Messwerten existiert (Abschnitt 7.4.3).

Zunächst wird die Planung speziell zur Berechnung von Ideallinien für Rundkurse entwickelt. Um die optimale Lösung zu erhalten, wird der Rundkurs als ganzes optimiert und nicht in Streckenabschnitte unterteilt. Dies führt zu einem leicht hochdimensionalen Optimierungsproblem (z. B. bei einem 4,5 km langen Kurs zu ca. 12 000 Optimierungsvariablen), bei dem i. A. keine gute Startlösung vorliegt, wofür das verwendete Interior-Point-Verfahren IPOPT prädestiniert ist. Ergebnisse der Offline-Planung zeigt Abschnitt 5.5.2. Im nächsten Schritt wird die Offline-Planung zu einer zyklischen Echtzeit-Planung mit gleitendem Horizont erweitert, welche auf sich verändernde Umgebungsbedingungen in Form von statischen Hindernissen reagieren kann. Mit ihr können Ausweichmanöver geplant werden, die so plötzlich erfolgen müssen, dass sie bis ans fahrdynamische Limit gehen. Die Planung liefert dabei automatisch eine optimale Kombination aus Verzögern und Ausweichen, wie die Ergebnisse in Abschnitt 7.4.1 zeigen.

Um das Optimierungsproblem besonders effizient lösen zu können, wurden alle benötigten Ableitungen der LAGRANGE-Funktion (Gradient, JACOBI- und HESSE-Matrix) analytisch berechnet. Um sie einerseits bestmöglich zusammenzufassen und zu vereinfachen und andererseits besonders effizient unter Ausnutzung von Leistungsmerkmalen der Zielhardware, wie Vektorisierung, zur Laufzeit zu berechnen, wurde auf den Einsatz von AD-Tools (automatisches Differenzieren) und automatischer Code-Generierung verzichtet.

Die vielen Arbeitsschritte zur Steigerung der Effizienz führen dazu, dass trotz der hohen Komplexität und Modellgenauigkeit extrem kurze Rechenzeiten erzielt werden (Rechenzeiten sind auf den Seiten 24, 89, 122 und 138 angegeben). Sie sind deutlich kürzer als in der Literatur zu findende Zeiten vergleichbarer Trajektorienplanungen und kürzer als viele Ansätze, die drastische Vereinfachungen, wie lineare Modelle oder eine geometrische Berechnung der Bahn, verwenden. Die vorliegende Arbeit zeigt, dass eine nichtlineare, modellbasierte numerische Optimierung, die mehrheitlich als nicht echtzeitfähig eingestuft wird, trotz verhältnismäßig langem Planungshorizont auf heutigen üblichen Mobile-CPU's und auch auf Embedded-PC's im Versuchsträger sogar dann in Echtzeit ausführbar ist, wenn sich das Fahrzeug mit hoher Geschwindigkeit bewegt.

Neben der eigentlichen Trajektorienplanung wurde zum einen ein Algorithmus entwickelt, um aus den per GPS gemessenen Fahrbahnrandern eine Referenzlinie für die Planung zu generieren, und zum anderen ein sogenannter Matching-Algorithmus, über den derjenige Punkt auf einem kubischen Spline berechnet wird, dessen Lot durch einen anderen vorgegebenen Punkt verläuft. Er wird verwendet, um zu einer kartesischen Position des Ego-Fahrzeugs die entsprechende Stelle entlang der Referenzlinie zu ermitteln.

Für die Planung ist es vorteilhaft, wenn die Referenzlinie möglichst glatt ist. Zu diesem Zweck wird ein Filter-Algorithmus entworfen, der mithilfe eines quadratischen Optimierungsproblems den Betrag der Krümmung sowie der ersten beiden Ableitungen der Krümmung von einer über die Weglänge parametrisierten Kurve im  $\mathbb{R}^2$  minimiert (Abschnitt 6.2). Optional können zusätzlich die Weglänge und die orthogonale Verschiebung zur Ausgangslinie minimiert werden. Mit oftmals geringfügigen Verschiebungen der Stützstellen gelingt es dem sogenannten Kappa-Filter, eine Kurve zu generieren, deren Krümmungsverlauf zweifach stetig differenzierbar ist. Er ähnelt im Ergebnis einer phasenfreien Tiefpass-Filterung, hat aber den Vorteil, einen Schlauch festle-

gen zu können, in dem die resultierende Kurve verlaufen muss, und über Gewichtungparameter die Form beeinflussen zu können. Der Algorithmus eignet sich bei entsprechender Parametrierung auch zur Berechnung einer krümmungsminimalen Linie als Approximation der Ideallinie. Die vorliegende Arbeit enthält dazu Gegenüberstellungen der Ergebnisse mit vergleichbaren Algorithmen aus der Literatur und mit der zeitoptimalen Planung (Abschnitt 6.2.4).

Auch wenn die Planung primär unter dem Aspekt der Zeitoptimalität entwickelt wurde, so ist die langfristige Zielsetzung eine allgemeine Trajektorienplanung, die das Potential hat, bis ans fahrdynamische Limit zu planen. Die vorliegende Arbeit zeigt im Rahmen einer Erweiterung zur „Komfort-Planung“, wie hohe und v. a. kombinierte Quer- und Längsbeschleunigungen in unkritischen Situationen vermieden werden können (Abschnitte 5.6 und 7.4.2). So ergeben sich komfortablere Trajektorien, die je nach Parametrierung unterschiedliche Geschwindigkeits- bzw. Beschleunigungsniveaus aufweisen, aber in jedem Fall angenehm glatt sind.

## Ausblick

Auf Basis der vorliegenden Arbeit bieten sich weitere Forschungs- und Entwicklungsthemen an, die im Folgenden umrissen werden.

Das wichtigste und wahrscheinlich umfangreichste Thema ist die Behandlung dynamischer Objekte. Während statische Objekte aufgrund der Ortsparametrierung ohne strukturelle Änderungen eingebaut werden können, erfordern dynamische Objekte eine *explizite* Betrachtung der Zeitverläufe des Ego-Fahrzeugs sowie des Objekts. Dabei ist die Zeit – wie alle Zustände – wegababhängig, also  $t(s)$ , was konsequenterweise nicht nur für das Ego-Fahrzeug, sondern auch für das Objekt gilt. Auch wenn die Darstellung  $t(s)$  recht ungewöhnlich ist, beinhaltet sie dieselbe Information wie  $s(t)$  und lässt sich für *bewegte* Objekte stets angeben, da die Funktion dann streng monoton ist (statische Objekte werden separat behandelt). Die Ego-Fahrzeit ergibt sich für jeden Abtastpunkt (analog der Berechnung der Gesamtfahrzeit im Gütefunktional) über

$$t_{\text{ego},k} = \sum_{i=1}^k h_i \cdot \sigma_i \quad \text{mit} \quad \sigma = \frac{1 - \kappa d}{v \cos(\psi_t + \beta)}. \quad (8.1)$$

Will man sie explizit im Optimierungsproblem verwenden, gibt es zwei Möglichkeiten:

- Bei jeder Verwendung von  $t_{\text{ego},k}$  die Summe (8.1) einsetzen, was allerdings dazu führt, dass die Anzahl der Abhängigkeiten entlang des Planungshorizonts mit jedem Diskretisierungspunkt um 4 Optimierungsvariablen steigt.
- Gleichung (8.1) als GNB formulieren und für  $t_{\text{ego},k}$   $N$  zusätzliche Optimierungsvariablen einführen. Allerdings ist diese Struktur für die Konvergenz nicht so schön, da eine Variation am Anfang des Horizonts durch alle GNB „durchrieseln“ muss, bis sie am Ende ankommt.

Für die Verwendung in einer Gütefunktion ist sicher die erste Möglichkeit geeigneter, da sie sich weder im Gradienten noch in der HESSE-Matrix negativ auf die Struktur des Optimierungsproblems auswirkt. Hingegen führt sie in einer NB zu mehr Einträgen in der JACOBI-Matrix. Generell

muss hierfür eine effiziente Methode entwickelt werden, wobei sicher auch gewisse Approximationen sinnvoll sind. In Anbetracht dessen, dass Position und Zeitpunkt des Objekts i. d. R. auf einer mehr oder weniger wagen Prädiktion beruhen, muss man sich der Frage stellen, wie groß der Nutzen ist,  $t_{\text{ego}}$  exakt zu berechnen. Besonders im realen Straßenverkehr dürfte eine Approximation  $\sigma \approx 1/v$ , wie in (5.10) auf S. 70, sinnvoll sein. Außerdem sind Güdeterme und/oder NB zu entwickeln, die Kollisionsfreiheit und entsprechende Manöver, wie Folgefahrt und Überholen, ermöglichen. Dabei sollte die Anzahl der Objekte keinen Einfluss auf die Struktur des Optimierungsproblems haben. Eine Schwierigkeit besteht darin, dass sich bei einem Überholmanöver i. A. mehrere Möglichkeiten, also mehrere lokale Optima, ergeben. Erste Untersuchungen und Entwicklungen zu dynamischen Objekten wurden im Rahmen des Projekts in [103] durchgeführt.

In Hinblick auf den realen Straßenverkehr benötigt diese Planung eine vorgeschaltete Taktik- und Manöverplanung, die bspw. den Fahrstreifen vorgibt oder ob und auf welcher Seite ein anderes Fahrzeug überholt werden soll. Diese Information kann in Form eines Fahrschlauchs übergeben werden, innerhalb dessen die fahrdynamisch berechnete Trajektorie liegen soll.

Der größte Nachteil einer nichtlinearen numerischen Optimierung ist, dass die Konvergenz nicht garantiert ist, geschweige denn die Rechenzeit bekannt oder gar begrenzt ist. Das heißt, sie ist nicht echtzeitfähig. In Abschnitt 7.1 wurden harte und weiche Echtzeitanforderungen an die Planung gestellt und dargelegt, dass die Planung abgebrochen werden muss und das Ergebnis nicht verwendet werden kann, wenn die harte Echtzeitanforderung verletzt wird, weil die Optimierung länger als prädiziert gedauert hat. Eine neue Planung auszuführen dauert länger, als wenn ihr von vornherein mehr Zeit gelassen worden wäre. Das Problem ist, dass bei einer zu pessimistischen Prädiktion die Agilität einer schnellen Planung nicht genutzt werden kann. Um sowohl kurze Rechenzeiten ausnutzen als auch Verletzungen der Echtzeitanforderung vermeiden zu können, können parallel zwei Planer mit unterschiedlicher Rechenzeitanforderung gestartet werden. Konvergiert der schnellere, wird der langsamere abgebrochen. Dennoch ist auch eine Backup-Planung (Redundanz) erforderlich, die auf einfachen Algorithmen beruht, um garantiert eine Notfall-Trajektorie zur Verfügung stellen zu können.

Für den realen Straßenverkehr ist eine Strategie im Fall einer unvermeidbaren Kollision notwendig. Sie kann entweder nachgelagert an diese Planung oder innerhalb dieser Planung umgesetzt werden. Im zweiten Fall müssen harte NB durch weiche Güdeterme ersetzt werden, damit mathematisch eine Lösung existiert. In die Strategie sollten Ergebnisse der Unfallforschung einfließen, die dann mathematisch in einem Optimierungsproblem umgesetzt werden.

In der Erweiterung zur „Komfort-Planung“ (Abschnitt 5.6) sowie in den sekundären Güdemaßen (Abschnitt 5.2.2) wurden bereits Komfortkriterien in der Planung berücksichtigt und es hat sich gezeigt, dass die Planung gut geeignet ist, um unterschiedliche Charakteristika abzubilden. Diese können mit zusätzlichen Kriterien, die das subjektive Komfortempfinden verbessern oder auch Kinetose (Bewegungskrankheit) vorbeugen, erweitert werden. Es wäre dabei zu untersuchen, ob alle relevanten Größen durch das vorhandene Modell abgedeckt werden können, oder ob bspw. die Wankdynamik modelliert werden sollte. Aus den Erkenntnissen ließen sich verschiedene Fahrprofile ableiten, die über Fahrmodi ausgewählt werden können.

## A Gleitkommazahlen nach IEEE 754

Die Repräsentation einer Gleitkommazahl  $x$  erfolgt bei allen gängigen Digitalrechnern nach dem IEEE 754-Standard [43] als Exponentialzahl zur Basis  $b = 2$  in der Form

$$x = (-1)^s m b^e. \quad (\text{A.1})$$

Eine 64 Bit-Zahl (double) hat 1 Vorzeichen-Bit  $s$ , 52 Bit für die Mantisse  $m$  und 11 Bit für den Exponenten  $e$ .

Genauso wie man im Dezimalsystem, also mit  $b = 10$ , jede Zahl so angeben kann, dass die führende Ziffer der Mantisse immer ungleich 0 ist, also z. B.  $00246 \cdot 10^0$  wird als  $24600 \cdot 10^{-2}$  angegeben, so kann man auch im Binärsystem jede Zahl ohne führende 0 angeben. Eine solche Zahl bezeichnet man als „normalisiert“. Da im Binärsystem nur die beiden Ziffern „0“ und „1“ existieren, ist die führende Ziffer einer normalisierten Binärzahl immer „1“. Diese führende binäre „1“ in der Mantisse ist somit implizit und wird nicht abgespeichert, sodass sich die Genauigkeit  $p = 53$  Bit ergibt, auch wenn nur 52 Bit abgespeichert werden.

Durch die endliche Anzahl an Bits in der Mantisse kann auch innerhalb der Grenzen des Definitionsbereichs nicht jede Zahl  $x \in \mathbb{R}$  dargestellt werden. Der Abstand zwischen einer (normalisierten) IEEE 754-Gleitkommazahl  $x$  und der nächstgrößeren beträgt mindestens  $b^{-1}\varepsilon|x|$  und maximal  $\varepsilon|x|$ , wobei  $\varepsilon = b^{1-p}$  die Maschinengenauigkeit ist [39, S. 37]. Die darstellbaren Zahlen sind also nicht äquidistant. Für eine Double-Zahl ergibt sich im ungünstigsten Fall ein *relativer* Quantisierungsfehler von

$$\varepsilon_{64} = 2^{1-53} \approx 2,22 \cdot 10^{-16} \quad (\text{A.2})$$

und für eine Single-Zahl sogar

$$\varepsilon_{32} = 2^{1-24} \approx 1,19 \cdot 10^{-7}. \quad (\text{A.3})$$

Für den Definitionsbereich von Gleitkommazahlen nach IEEE 754 sowie die Darstellung denormalisierter Zahlen und Sonderwerte, wie Unendlich und NaN (not a number), sei auf die angegebene Literatur [43, 39] verwiesen.

## B Autobahnquerschnitte und Kurvenradien

Um bezüglich der Approximation (5.10) auf S. 70 eine Abschätzung zu treffen, wie groß der durch die Annahme  $\kappa d \approx 0$  entstehende Fehler ist, werden in der folgenden Tabelle Maximalwerte für die Krümmung sowie die Fahrbahnbreite auf deutschen Autobahnen angegeben. Hierfür werden die *Richtlinien für die Anlage von Autobahnen* [68] herangezogen. Der maximale Wert für  $\kappa d$  ergibt sich unter der Annahme, dass die Referenzlinie in der Fahrbahnmitte liegt. Der tatsächliche Maximalwert für  $d$  ist geringer als angegeben, da die Fzg-Breite von  $d_{\text{Fb}}$  abzuziehen ist.

Entwurfsklasse	EKA 1			EKA 2	EKA 3		
Netzfunktion	Fern- u. überregionale Autobahnen			AB-ähnliche Straßen	Stadtautobahnen mit $v_{\text{max}} \leq 100 \text{ km/h}$		
Regelquerschnitt	RQ 43,5	RQ 36	RQ 31	RQ 28	RQ 38,5	RQ 31,5	RQ 25
Kurvenmindest-radius $r_{\text{min}}$ in m	900			470	280		
Fahrbahnbreite pro Richtung $d_{\text{Fb}}$ in m	$2 \cdot 3,5 + 2 \cdot 3,75 = 14,5$	$2 \cdot 3,5 + 3,75 = 10,75$	$2 \cdot 3,75 = 7,5$	$2 \cdot 3,5 = 7$	$2 \cdot 3,25 + 2 \cdot 3,5 = 13,5$	$2 \cdot 3,25 + 3,5 = 20$	$3,25 + 3,5 = 6,75$
$\kappa d < \frac{1}{2} d_{\text{Fb}} r_{\text{min}}^{-1} =$	0,0081	0,0060	0,0052	0,0074	<b>0,0241</b>	0,0179	0,0121

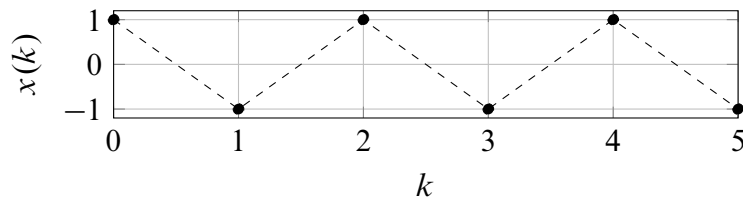


## C Differenzenquotient

In Abschnitt 5.1.3 wird ein Ansatz zur Berechnung der Koeffizienten eines Vorwärts-Differenzenquotienten höherer Ordnung beschrieben. Die Ableitung einer Funktion – diskret approximiert durch ihren Differenzenquotienten – wird dann quadriert und im Gütemaß bestraft, um im Ergebnis der Optimierung einen möglichst glatten Verlauf dieser Funktion zu erhalten. Dabei wird gesagt, dass hierfür ein zentraler Differenzenquotient ungeeignet ist, weil die Optimierung dann nicht verhindern kann, dass die Funktion mit der Abtastfrequenz schwingt. Irgendeine kleine Anregung an einer Stelle genügt, um diese Schwingung zu erzeugen. Das Gütemaß hat im Falle eines zentralen Differenzenquotienten für die schwingende Funktion denselben Wert, wie bei einer nicht schwingenden Funktion. Dieses Problem kann durch Verwendung eines Vorwärts- oder Rückwärts-Differenzenquotienten umgangen werden. Im Folgenden wird gezeigt, dass dieses Verhalten auf eine Übertragungsnullstelle im Differenzierer zurückzuführen ist.

Wir betrachten hierfür die diskrete für  $k \geq 0$  periodische Folge

$$x(k) = (-1)^k, \quad k \geq 0. \quad (\text{C.1})$$



Die z-Transformierte lautet somit:

$$X(z) = \mathcal{Z} \{x(k)\} = \sum_{k=0}^{\infty} (-1)^k z^{-k} = \sum_{k=0}^{\infty} \left(-\frac{1}{z}\right)^k = \frac{1}{1 - (-\frac{1}{z})} = \frac{z}{z+1} \quad (\text{C.2})$$

Betrachtet man  $x(k)$  als die abgetastete Folge einer für  $t \geq 0$  periodischen Funktion  $x(t)$ , kann man sehen, dass die Funktion gerade mit ihrer Abtastfrequenz schwingt. Die Polstelle bei  $z = -1$  entspricht im LAPLACE-Bereich mit der LAPLACE-Variablen  $s = \sigma + j\omega$  und der Abtastzeit  $T > 0$

$$-1 \stackrel{!}{=} z = e^{Ts} = \underbrace{e^{\sigma T}}_1 \underbrace{e^{j\omega T}}_{-1} \Rightarrow \begin{cases} \sigma T = 0, \\ \omega T = \pi, \end{cases} \quad (\text{C.3})$$

also einer ungedämpften Schwingung mit der Frequenz  $\omega = \frac{\pi}{T}$  oder auch

$$f = \frac{\omega}{2\pi} = \frac{1}{2} f_s, \quad (\text{C.4})$$

wobei  $f_s = \frac{1}{T}$  die Abtastfrequenz ist.

## C.1 Zentraler Differenzenquotient

Der zentrale Differenzenquotient der Folge  $x(k)$  sei  $y_z(k)$ . Er berechnet sich durch

$$y_z(k) = \frac{x(k+1) - x(k-1)}{2T}. \quad (\text{C.5})$$

Schon im Zeitbereich kann man sehen, dass  $y_z$  keine Information über die Schwingung beinhaltet, denn bis auf einen Einschwingvorgang ist die so berechnete Ableitung null:

$$x(k) = \begin{cases} 0, & k < 0 \\ (-1)^k, & k \geq 0 \end{cases} \Rightarrow y_z(k) = \frac{x(k+1) - x(k-1)}{2T} = \begin{cases} 0, & k \leq -2 \\ \frac{1}{2T}, & k = -1 \\ -\frac{1}{2T}, & k = 0 \\ 0, & k \geq 1 \end{cases} \quad (\text{C.6})$$

Die Übertragungsfunktion des zentralen Differenzenquotienten (C.5) ist

$$G_z(z) = \frac{Y_z(z)}{X(z)} = \frac{1}{2} \left( z - \frac{1}{z} \right) = \frac{1}{2} \frac{z^2 - 1}{z} = \frac{1}{2} \frac{(z-1)(z+1)}{z}. \quad (\text{C.7})$$

Sie ist natürlich nicht kausal, aber das ist für diese Anwendung auch nicht notwendig. Somit ergibt sich aus (C.7) und (C.2) das Ausgangssignal des Differenzierers:

$$Y_z(z) = G_z(z) X(z) = \frac{1}{2} \frac{(z-1)(z+1)}{z} \frac{z}{(z+1)} = \frac{1}{2} (z-1) \quad (\text{C.8})$$

$$\begin{array}{c} \bullet \\ | \\ \circ \end{array} \quad y_z^*(t) = \frac{1}{2} (\delta(t+T) - \delta(t)) \quad (\text{C.9})$$

In (C.8) sieht man, dass die Dynamik des Eingangssignals (C.2) aufgrund der Nullstelle von  $G_z(z)$  bei  $z = -1$  nicht übertragen wird.

Es kann also passieren, dass sich aus der Optimierung eine Funktion ergibt, die mit der halben Abtastfrequenz oszilliert, ohne dass diese Schwingung irgendeine Auswirkung auf das Gütemaß hat. Dies entspricht gerade der Grenzfrequenz des SHANNON-Theorems, welches besagt, dass die höchste im System vorkommende Frequenz mindestens doppelt so groß wie die Abtastfrequenz sein muss. Eine weitere Verletzung des SHANNON-Theorems durch noch höhere Frequenzen ist a priori ausgeschlossen, da in der Optimierung keine Werte zwischen den Abtastpunkten existieren.

## C.2 Vorwärts-Differenzenquotienten

Eine andere Situation ergibt sich beim Vorwärts-Differenzenquotienten (ebenso beim Rückwärts-Differenzenquotienten). Er berechnet sich durch

$$y_{vw}(k) = \frac{x(k+1) - x(k)}{T}, \quad (\text{C.10})$$

woraus sich die Übertragungsfunktion

$$G_{\text{vw}}(z) = z - 1 \quad (\text{C.11})$$

ergibt. Das Ausgangssignal ist dann:

$$Y_{\text{vw}}(z) = G_{\text{vw}}(z) X(z) = (z - 1) \frac{z}{z + 1} = z \frac{z}{z + 1} - \frac{z}{z + 1} \quad (\text{C.12})$$



$$y_{\text{vw}}(k) = (-1)^{k+1} - (-1)^k = -2 \cdot (-1)^k, \quad k \geq 0 \quad (\text{C.13})$$

Dies entspricht der Erwartung, dass  $y_{\text{vw}}(k)$  an jedem Punkt die Steigung der Interpolationsgeraden zwischen dem aktuellen und dem nachfolgenden Signalwert angibt.

## D Ableitungen im Optimierungsproblem

Im Folgenden sind die Ableitungen, die zur Berechnung der *evaluation functions* benötigt werden, nämlich der Gradient der Gütefunktion, die JACOBI-Matrix aller Nebenbedingungen sowie die HESSE-Matrix der gesamten LAGRANGE-Funktion aufgeführt. Zur Notation des Gradienten sei auf die Definition auf S. 8 verwiesen. Selbstverständlich werden die Funktionen nach dem Vektor der Optimierungsvariablen  $\mathbf{x}_{\text{opt}}$  abgeleitet, jedoch ist die Darstellung übersichtlicher, wenn man statt der einzelnen diskreten Optimierungsvariablen nur skalare Zustandsvariablen aufschreibt:

$$\nabla^\top f(\mathbf{x}_{\text{opt}}) = \begin{cases} \left[ \frac{\partial f(\mathbf{x}_{\text{opt}})}{\partial \mathbf{x}_{\text{opt},1}} & \dots & \frac{\partial f(\mathbf{x}_{\text{opt}})}{\partial \mathbf{x}_{\text{opt},8N}} \right], & \text{Rundkurs} \\ \left[ \frac{\partial f(\mathbf{x}_{\text{opt}})}{\partial \mathbf{x}_{\text{opt},1}} & \dots & \frac{\partial f(\mathbf{x}_{\text{opt}})}{\partial \mathbf{x}_{\text{opt},8N+5}} \right], & \text{offene Strecke} \end{cases} \quad (\text{D.1a})$$

$$\leadsto \nabla^\top f(\mathbf{x}, \mathbf{u}) = \left[ \frac{\partial f(\mathbf{x}, \mathbf{u})}{\partial u_M} \quad \frac{\partial f(\mathbf{x}, \mathbf{u})}{\partial u_B} \quad \frac{\partial f(\mathbf{x}, \mathbf{u})}{\partial \delta} \quad \frac{\partial f(\mathbf{x}, \mathbf{u})}{\partial v} \quad \frac{\partial f(\mathbf{x}, \mathbf{u})}{\partial d} \quad \frac{\partial f(\mathbf{x}, \mathbf{u})}{\partial \beta} \quad \frac{\partial f(\mathbf{x}, \mathbf{u})}{\partial \psi_t} \quad \frac{\partial f(\mathbf{x}, \mathbf{u})}{\partial \dot{\psi}} \right] \quad (\text{D.1b})$$

Alle Funktionen weisen dieselbe Abhängigkeit von  $\mathbf{x}_{\text{opt}}$  bzw.  $\mathbf{x}, \mathbf{u}$  auf, sodass sie im Folgenden weggelassen wird.

Da alle 2. Ableitungen stetig sind, ist die HESSE-Matrix nach dem Satz von SCHWARZ symmetrisch [16, S. 411]. Und da dies eine Voraussetzung in IPOPT ist, wird dort lediglich die untere Dreiecksmatrix verwendet. *Deswegen werden in allen folgenden 2. Ableitungen nur die Terme der unteren Dreiecksmatrix aufgeführt und alle anderen ignoriert.*

### D.1 Gradient und HESSE-Matrix der Gütefunktion

Der Gradient von (5.23) ist

$$\nabla J_t = \sum_{k=1}^N h_k \nabla \sigma_k, \quad (\text{D.2})$$

wobei sich für einen einzelnen Summanden folgende Terme ergeben:

$$\frac{\partial \sigma}{\partial u_M} = \frac{\partial \sigma}{\partial u_B} = \frac{\partial \sigma}{\partial \delta} = \frac{\partial \sigma}{\partial \dot{\psi}} = 0 \quad (\text{D.3a})$$

$$\frac{\partial \sigma}{\partial v} = -\frac{\sigma}{v} \quad (\text{D.3b})$$

$$\frac{\partial \sigma}{\partial d} = -\frac{\kappa}{v \cos(\psi_t + \beta)} \quad (\text{D.3c})$$

$$\frac{\partial \sigma}{\partial \beta} = \frac{\partial \sigma}{\partial \psi_t} = \sigma \tan(\psi_t + \beta) \quad (\text{D.3d})$$

Die 2. Ableitungen, die ungleich null sind, ergeben sich zu:

$$\frac{\partial^2 \sigma}{\partial v^2} = 2 \frac{\sigma}{v^2} \quad (\text{D.4a})$$

$$\frac{\partial^2 \sigma}{\partial d \partial v} = \frac{\kappa}{v^2 \cos(\psi_t + \beta)} \quad (\text{D.4b})$$

$$\frac{\partial^2 \sigma}{\partial \beta \partial v} = \frac{\partial^2 \sigma}{\partial \psi_t \partial v} = -\frac{\sigma}{v} \tan(\psi_t + \beta) \quad (\text{D.4c})$$

$$\frac{\partial^2 \sigma}{\partial \beta \partial d} = \frac{\partial^2 \sigma}{\partial \psi_t \partial d} = -\frac{\kappa}{v \cos(\psi_t + \beta)} \tan(\psi_t + \beta) \quad (\text{D.4d})$$

$$\frac{\partial^2 \sigma}{\partial \beta^2} = \frac{\partial^2 \sigma}{\partial \psi_t \partial \beta} = \frac{\partial^2 \sigma}{\partial \psi_t^2} = \sigma \left( \frac{2}{\cos(\psi_t + \beta)^2} - 1 \right) \quad (\text{D.4e})$$

Die Ableitungen der Straffunktion (5.24) sind:

$$\nabla \Phi_{\text{MBK}} = \lambda_{\text{MBK}} u_{\text{M}} u_{\text{B}} \nabla(u_{\text{M}} u_{\text{B}}), \quad \nabla(u_{\text{M}} u_{\text{B}}) = [u_{\text{B}} \quad u_{\text{M}} \quad 0 \quad 0 \quad \dots \quad 0] \quad (\text{D.5})$$

$$\nabla^2 \Phi_{\text{MBK}} = \lambda_{\text{MBK}} \begin{bmatrix} u_{\text{B}}^2 & & & \\ 2u_{\text{M}}u_{\text{B}} & u_{\text{M}}^2 & & \\ 0 & 0 & 0 & \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (\text{D.6})$$

Die Ableitungen der Gütefunktionen (5.26), (5.27), (5.33) und (5.34) ergeben sich durch Anwendung von (5.22).

## D.2 JACOBI-Matrix der Nebenbedingungen

### D.2.1 Systemgleichungen

Die folgenden Ableitungen der Systemgleichungen aus (5.39) ergeben sich durch Differentiation von (3.34a), (3.7), (3.34b), (3.9) und (3.32c). Die dabei entstehenden Ableitungen der Reifenkräfte werden in Abschnitt D.2.4 angegeben.

$$\begin{aligned} \nabla \dot{v} &= \frac{1}{m} \left( F_{x_{\text{rv}}} \sin(\beta - \delta) - F_{y_{\text{rv}}} \cos(\beta - \delta) \right) \nabla \delta + v(\dot{\beta} + \dot{\psi}) \nabla \beta + \dots \\ &\quad \frac{1}{m} \left( \cos(\beta - \delta) \nabla F_{x_{\text{rv}}} + \cos(\beta) \nabla (F_{x_{\text{rh}}} - F_{w,x}) \right) + \dots \\ &\quad \frac{1}{m} \left( \sin(\beta - \delta) \nabla F_{y_{\text{rv}}} + \sin(\beta) \nabla F_{y_{\text{rh}}} \right) \end{aligned} \quad (\text{D.7})$$

$$\nabla \dot{d} = \sin(\psi_t + \beta) \nabla v + v \cos(\psi_t + \beta) \nabla(\psi_t + \beta) \quad (\text{D.8})$$

$$\begin{aligned}
\nabla \dot{\beta} &= -\frac{1}{v} (\dot{\beta} + \dot{\psi}) \nabla v - \frac{1}{v} \dot{v} \nabla \beta - \nabla \dot{\psi} + \dots \\
&+ \frac{1}{mv} (F_{x_r v} \cos(\beta - \delta) + F_{y_r v} \sin(\beta - \delta)) \nabla \delta + \dots \\
&+ \frac{1}{mv} (-\sin(\beta - \delta) \nabla F_{x_r v} - \sin(\beta) \nabla (F_{x_r h} - F_{w, x}) + \dots \\
&\quad + \cos(\beta - \delta) \nabla F_{y_r v} + \cos(\beta) \nabla F_{y_r h})
\end{aligned} \tag{D.9}$$

$$\nabla (\sigma \dot{\psi}) = \nabla \sigma \dot{\psi} + \sigma \nabla \dot{\psi} \tag{D.10}$$

$$\begin{aligned}
\nabla \ddot{\psi} &= \frac{l_v}{J_z} (\sin(\delta) \nabla F_{x_r v} + F_{x_r v} \cos(\delta) \nabla \delta + \dots \\
&\quad + \cos(\delta) \nabla F_{y_r v} - F_{y_r v} \sin(\delta) \nabla \delta) - \frac{l_h}{J_z} \nabla F_{y_r h}
\end{aligned} \tag{D.11}$$

### D.2.2 Kammscher Kreis

Der Gradient von (5.42) ist

$$\begin{aligned}
\nabla F_{\text{norm}}^2 &= 2 \frac{F_{x_r}}{F_{\text{max}, x_r}^2} \left( \frac{\partial F_{x_r}}{\partial F_M} \pm \mu_x \frac{h_{\text{SP}}}{l} \frac{F_{x_r}}{F_{\text{max}, x_r}} \right) \nabla F_M + \dots \\
&\quad + 2 \frac{F_{x_r}}{F_{\text{max}, x_r}^2} \left( \frac{\partial F_{x_r}}{\partial F_B} \pm \mu_x \frac{h_{\text{SP}}}{l} \frac{F_{x_r}}{F_{\text{max}, x_r}} \right) \nabla F_B + 2 F_{y_r, \text{rel}} \nabla F_{y_r, \text{rel}}.
\end{aligned} \tag{D.12}$$

Darin gilt die obere Lesart bei „ $\pm$ “ für die Vorderachse, die untere für die Hinterachse.  $\nabla F_{y_r, \text{rel}}$  ist in Abschnitt D.2.4 angegeben.

### D.2.3 Maximale Längskraft durch die Motorleistung

Der Gradient von (5.43) ist

$$\nabla (F_M v) = F_M \nabla v + v \nabla F_M. \tag{D.13}$$

### D.2.4 Reifenkräfte und Schräglaufwinkel

Aus (3.57) sowie der Widerstandskraft (3.42) folgen die Ableitungen der Längskräfte:

$$\nabla F_{x_r v} = k_{\text{MV}} \nabla F_M + (k_{\text{BV}, 0} - 2 k_{\text{BV}, p} F_B) \nabla F_B \tag{D.14a}$$

$$\nabla F_{x_r h} = (1 - k_{\text{MV}}) \nabla F_M + (1 - k_{\text{BV}, 0} + 2 k_{\text{BV}, p} F_B) \nabla F_B \tag{D.14b}$$

$$\nabla (F_{x_r h} - F_{w, x}) = \nabla F_{x_r h} - 2 k_{\text{Lx}} v \nabla v \tag{D.14c}$$

Die folgenden Ableitungen der Querkräfte gelten sowohl für die Vorder- als auch die Hinterachse.

$$\nabla F_{y_r} = F_{\max, y_r} \nabla F_{y_r, \text{rel}} + \underbrace{F_{y_r, \text{rel}} \nabla F_{\max, y_r}}_{\text{nur für } h_{\text{SP}} > 0} \quad (\text{D.15a})$$

$$\text{mit } \nabla F_{y_r, \text{rel}} = BC \frac{\cos(C \arctan(B\alpha))}{1 + (B\alpha)^2} \nabla \alpha, \quad (\text{D.15b})$$

$$\nabla^\top \alpha_v = \begin{bmatrix} 0 & 0 & 1 & +\frac{\dot{\psi} l_v}{v^2} & 0 & -1 & 0 & -\frac{l_v}{v} \end{bmatrix}, \quad (\text{D.15c})$$

$$\nabla^\top \alpha_h = \begin{bmatrix} 0 & 0 & 0 & -\frac{\dot{\psi} l_h}{v^2} & 0 & -1 & 0 & +\frac{l_h}{v} \end{bmatrix}, \quad (\text{D.15d})$$

$$\nabla F_{\max, y_r} = \mp \mu_y \frac{h_{\text{SP}}}{l} \nabla (F_M + F_B) \quad (, - \text{“ für vorne}) \quad (\text{D.15e})$$

## D.3 HESSE-Matrizen der Nebenbedingungen

### D.3.1 Systemgleichungen

In diesem Abschnitt werden alle noch nicht berechneten Terme aus (5.41) angegeben:

$$\begin{aligned} \nabla^2 \dot{v} &= \frac{1}{m} (F_{x_rv} \cos(\beta - \delta) + F_{y_rv} \sin(\beta - \delta)) (\nabla \beta \nabla^\top \delta - \nabla \delta \nabla^\top \beta) + \dots \\ &\quad - \dot{v} \nabla \beta \nabla^\top \beta - \frac{1}{m} \sin(\beta - \delta) \nabla(\beta - \delta) \nabla^\top F_{x_rv} + \dots \\ &\quad - \frac{1}{m} \sin(\beta) \nabla \beta \nabla^\top (F_{x_{rh}} - F_{w,x}) + \dots \\ &\quad + \frac{1}{m} \cos(\beta - \delta) (\nabla F_{y_rv} \nabla^\top (\beta - \delta) + \nabla(\beta - \delta) \nabla^\top F_{y_rv}) + \dots \\ &\quad + \frac{1}{m} \cos(\beta) (\nabla F_{y_{rh}} \nabla^\top \beta + \nabla \beta \nabla^\top F_{y_{rh}}) + \dots \\ &\quad + \frac{2k_{\text{BV,p}}}{m} (\cos(\beta) - \cos(\beta - \delta)) \nabla F_B \nabla^\top F_B + \dots \\ &\quad - \frac{1}{m} \cos(\beta) \nabla^2 F_{w,x} + \frac{1}{m} \sin(\beta - \delta) \nabla^2 F_{y_rv} + \frac{1}{m} \sin(\beta) \nabla^2 F_{y_{rh}} \end{aligned} \quad (\text{D.16})$$

$$\begin{aligned} \nabla^2 f_d &= -\frac{h\kappa}{\cos(\psi_t + \beta)^2} (\nabla(\beta + \psi_t) \nabla^\top d) + \dots \\ &\quad + 2h(1 - \kappa d) \frac{\tan(\psi_t + \beta)}{\cos(\psi_t + \beta)^2} (\nabla \beta \nabla^\top \beta + \nabla \psi_t \nabla^\top \beta + \nabla \psi_t \nabla^\top \psi_t) \end{aligned} \quad (\text{D.17})$$

$$\begin{aligned}
\nabla^2 \dot{\beta} = & \frac{2}{v^2} (\dot{\beta} + \dot{\psi}) \nabla v \nabla^\top v - (\dot{\beta} + \dot{\psi}) \nabla \beta \nabla^\top \beta + \frac{1}{v^2} \dot{v} \nabla \beta \nabla^\top v + \dots \\
& + \frac{1}{mv^2} \nabla v \left( (\cos(\beta - \delta) F_{x_r v} + \sin(\beta - \delta) F_{y_r v}) \nabla^\top (-\delta) \right) + \dots \\
& + \frac{\sin(\beta - \delta)}{mv^2} \nabla v \nabla^\top F_{x_r v} + \dots \\
& + \frac{\sin(\beta)}{mv^2} \left( \nabla v \nabla^\top (F_{x_r h} - F_{w, x}) + \nabla (F_{x_r h} - F_{w, x}) \nabla^\top v \right) + \dots \\
& - \frac{\cos(\beta - \delta)}{mv^2} \left( \nabla v \nabla^\top F_{y_r v} + \nabla F_{y_r v} \nabla^\top v \right) + \dots \\
& - \frac{\cos(\beta)}{mv^2} \left( \nabla v \nabla^\top F_{y_r h} + \nabla F_{y_r h} \nabla^\top v \right) + \dots \\
& - \frac{\cos(\beta - \delta)}{mv} \nabla (\beta - \delta) \nabla^\top F_{x_r v} - \frac{\cos(\beta)}{mv} \nabla \beta \nabla^\top (F_{x_r h} - F_{w, x}) + \dots \\
& + \frac{1}{mv} \left( \sin(\beta - \delta) F_{x_r v} - \cos(\beta - \delta) F_{y_r v} \right) (\nabla \delta \nabla^\top \delta - \nabla \beta \nabla^\top \delta) + \dots \\
& - \frac{\sin(\beta - \delta)}{mv} \left( \nabla F_{y_r v} \nabla^\top (\beta - \delta) + \nabla (\beta - \delta) \nabla^\top F_{y_r v} \right) + \dots \\
& + \frac{2k_{BV,p}}{mv} \left( \sin(\beta - \delta) - \sin(\beta) \right) \nabla F_B \nabla^\top F_B + \frac{\sin(\beta)}{mv} \nabla^2 F_{w, x} + \dots \\
& - \frac{\sin(\beta)}{mv} \left( \nabla F_{y_r h} \nabla^\top \beta + \nabla \beta \nabla^\top F_{y_r h} \right) + \frac{\cos(\beta - \delta)}{mv} \nabla^2 F_{y_r v} + \frac{\cos(\beta)}{mv} \nabla^2 F_{y_r h}
\end{aligned} \tag{D.18}$$

$$\begin{aligned}
\nabla^2 \ddot{\psi} = & \frac{l_v}{J_z} \left( \cos(\delta) \nabla \delta \nabla^\top F_{x_r v} - (\sin(\delta) F_{x_r v} + \cos(\delta) F_{y_r v}) \nabla \delta \nabla^\top \delta + \dots \right. \\
& \left. - \sin(\delta) \left( \nabla \delta \nabla^\top F_{y_r v} + \nabla F_{y_r v} \nabla^\top \delta \right) + \dots \right. \\
& \left. + \sin(\delta) \nabla^2 F_{x_r v} + \cos(\delta) \nabla^2 F_{y_r v} \right) - \frac{l_h}{J_z} \nabla^2 F_{y_r h}
\end{aligned} \tag{D.19}$$



### D.3.2 Kammscher Kreis

Die HESSE-Matrix von (5.42) ist

$$\begin{aligned} \nabla^2 F_{\text{norm}}^2 = & 2 \left( \frac{1}{F_{\text{max},x_r}^2} (\nabla F_{x_r} \nabla^\top F_{x_r} + F_{x_r} \nabla^2 F_{x_r}) + \dots \right. \\ & + \nabla F_{y_r, \text{rel}} \nabla^\top F_{y_r, \text{rel}} + F_{y_r, \text{rel}} \nabla^2 F_{y_r, \text{rel}} + \dots \\ & \left. - \underbrace{\frac{2F_{x_r}}{F_{\text{max},x_r}^3} (\nabla F_{x_r} \nabla^\top F_{\text{max},x_r} + \nabla F_{\text{max},x_r} \nabla^\top F_{x_r}) + \frac{3F_{x_r}^2}{F_{\text{max},x_r}^4} \nabla F_{\text{max},x_r} \nabla^\top F_{\text{max},x_r}}_{\text{nur für } h_{\text{SP}} > 0} \right). \end{aligned} \quad (\text{D.20})$$

Darin ist

$$\begin{aligned} & \nabla F_{y_r, \text{rel}} \nabla^\top F_{y_r, \text{rel}} + F_{y_r, \text{rel}} \nabla^2 F_{y_r, \text{rel}} \\ &= \left( \frac{\partial F_{y_r, \text{rel}}}{\partial \alpha} \right)^2 \nabla \alpha \nabla^\top \alpha + F_{y_r, \text{rel}} \left( \frac{\partial F_{y_r, \text{rel}}}{\partial \alpha} \nabla^2 \alpha + \frac{\partial^2 F_{y_r, \text{rel}}}{\partial \alpha^2} \nabla \alpha \nabla^\top \alpha \right) \\ &= F_{y_r, \text{rel}} \frac{\partial F_{y_r, \text{rel}}}{\partial \alpha} \nabla^2 \alpha + \left( \left( \frac{\partial F_{y_r, \text{rel}}}{\partial \alpha} \right)^2 + F_{y_r, \text{rel}} \frac{\partial^2 F_{y_r, \text{rel}}}{\partial \alpha^2} \right) \nabla \alpha \nabla^\top \alpha. \end{aligned} \quad (\text{D.21})$$

Die partiellen Ableitungen der Querkraft und die Gradienten der Schräglaufwinkel sind in Abschnitt D.3.4 aufgeführt.

### D.3.3 Maximale Längskraft durch die Motorleistung

Die HESSE-Matrix von (5.43) ist

$$\nabla^2 (F_M v) = \nabla v \nabla^\top F_M. \quad (\text{D.22})$$

### D.3.4 Reifenkräfte und Schräglaufwinkel

Die HESSE-Matrizen der Längskräfte  $F_{x_r v}$  und  $F_{x_r h}$  sind nur für eine dynamische Bremskraftverteilung  $k_{\text{BV}} \neq \text{const}$  ungleich Null.

$$\nabla^2 F_{x_r v} = -2 k_{\text{BV}, p} \nabla F_B \nabla^\top F_B \quad (\text{D.23a})$$

$$\nabla^2 F_{x_r h} = +2 k_{\text{BV}, p} \nabla F_B \nabla^\top F_B \quad (\text{D.23b})$$

Die HESSE-Matrix der Querkräfte  $F_{y_r(v/h)}$ , welche in den Gleichungen in Abschnitt D.3.1 und D.3.2 bereits mehrmals verwendet wurden, ergeben sich aus (D.15) für vorne bzw. hinten durch:

$$\begin{aligned} \nabla^2 F_{y_r} = & F_{\max, y_r} \frac{\partial F_{y_r, \text{rel}}}{\partial \alpha} \nabla^2 \alpha + F_{\max, y_r} \frac{\partial^2 F_{y_r, \text{rel}}}{\partial \alpha^2} \nabla \alpha \nabla^\top \alpha + \dots \\ & + \underbrace{\frac{\partial F_{y_r, \text{rel}}}{\partial \alpha} \frac{\partial F_{\max, y_r}}{\partial F_{m/b}} \nabla \alpha \nabla^\top (F_M + F_B)}_{\text{nur für } h_{\text{SP}} > 0} \end{aligned} \quad (\text{D.24})$$

Die Ableitungen der Querkraft nach dem Schräglaufwinkel für vorne bzw. hinten sind

$$\frac{\partial F_{y_r, \text{rel}}}{\partial \alpha} = \frac{BC}{1 + (B\alpha)^2} \cos(C \arctan(B\alpha)) \quad (\text{D.25a})$$

$$\frac{\partial^2 F_{y_r, \text{rel}}}{\partial \alpha^2} = -B \frac{BC}{1 + (B\alpha)^2} \frac{C \sin(C \arctan(B\alpha)) + 2B\alpha \cos(C \arctan(B\alpha))}{1 + (B\alpha)^2} \quad (\text{D.25b})$$

und die Produkte der Gradienten der Schräglaufwinkel sind:

$$\nabla \alpha_v \nabla^\top \alpha_v = \begin{bmatrix} 0 \\ 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & \frac{\dot{\psi} l_v}{v^2} & \left(\frac{\dot{\psi} l_v}{v^2}\right)^2 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -\frac{\dot{\psi} l_v}{v^2} & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{l_v}{v} & -\frac{\dot{\psi} l_v^2}{v^3} & 0 & \frac{l_v}{v} & 0 & \left(\frac{l_v}{v}\right)^2 \end{bmatrix} \quad (\text{D.26a})$$

$$\nabla \alpha_h \nabla^\top \alpha_h = \begin{bmatrix} 0 \\ 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 & \left(\frac{\dot{\psi} l_h}{v^2}\right)^2 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\dot{\psi} l_h}{v^2} & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{\dot{\psi} l_h^2}{v^3} & 0 & -\frac{l_h}{v} & 0 & \left(\frac{l_h}{v}\right)^2 \end{bmatrix} \quad (\text{D.26b})$$

Die HESSE-Matrizen der Schräglaufwinkel enthalten vorne und hinten jeweils nur zwei Elemente:

$$\frac{\partial^2 \alpha_v}{\partial v^2} = -2 \frac{\dot{\psi} l_v}{v^3}, \quad \frac{\partial^2 \alpha_v}{\partial \dot{\psi} \partial v} = + \frac{l_v}{v^2} \quad (\text{D.27a})$$

$$\frac{\partial^2 \alpha_h}{\partial v^2} = +2 \frac{\dot{\psi} l_h}{v^3}, \quad \frac{\partial^2 \alpha_h}{\partial \dot{\psi} \partial v} = - \frac{l_h}{v^2} \quad (\text{D.27b})$$

## E Box-Restriktion der Längskraft

Eine Begrenzung der Längskräfte ist nur durch  $F_M \geq 0$  und  $F_B \leq 0$  notwendig, weil die jeweils andere Richtung durch die nichtlinearen NB für den Kammschen Kreis und das Lieferkennfeld begrenzt wird. Allerdings lassen sich Box-Restriktionen im Optimierungsproblem viel leichter einhalten als nichtlineare NB, und die Effizienz kann steigen, wenn der Bereich einzelner Optimierungsvariablen zusätzlich eingeschränkt ist. Da die Grenze konstant sein muss, also keine Variablenabhängigkeit aufweisen darf, können nur allgemeine obere bzw. untere Grenzen

$$0 \leq F_M \leq F_{M,\max} \quad (\text{E.1a})$$

$$F_{B,\min} \leq F_B \leq 0 \quad (\text{E.1b})$$

bei  $F_{y_r} = 0$  angesetzt werden. Sie hängen allerdings von der Kraftverteilung auf die Achsen ab:

$$F_{M,\max} = \min \left\{ \frac{F_{\max,xrv}}{k_{MV}}, \frac{F_{\max,xrh}}{1 - k_{MV}} \right\} = \mu_x \min \left\{ \frac{F_{zrv}}{k_{MV}}, \frac{F_{zrh}}{1 - k_{MV}} \right\} \Big|_{F_B=0} \quad (\text{E.2a})$$

$$F_{B,\min} = -\min \left\{ \frac{F_{\max,xrv}}{k_{BV}}, \frac{F_{\max,xrh}}{1 - k_{BV}} \right\} = -\mu_x \min \left\{ \frac{F_{zrv}}{k_{BV}}, \frac{F_{zrh}}{1 - k_{BV}} \right\} \Big|_{F_M=0} \quad (\text{E.2b})$$

Darin wird für  $F_{z_r}$  der jeweilige Term aus (3.51) auf S. 42 eingesetzt, welcher allerdings selbst von  $F_M$  und  $F_B$  abhängt. Da sich  $F_M$  und  $F_B$  ausschließen, kann davon ausgegangen werden, dass in (E.2a)  $F_B = 0$  und in (E.2b)  $F_M = 0$  ist.

Der Wert  $F_{B,\min}$  in (E.2b) lässt sich direkt ausrechnen, indem man (3.56) und (3.55) einsetzt:

$$\left. \begin{aligned} \frac{F_{zrv}}{k_{BV}} &= \frac{F_{zrv}}{\hat{F}_{Bv}} = \frac{F_{zrv}}{\mu_x F_{zrv}} = G \\ \frac{F_{zrh}}{1 - k_{BV}} &= \frac{F_{zrh}}{\hat{F}_{Bh}} = \frac{F_{zrh}}{\mu_x (F_{zrv} + F_{zrh})} = G \end{aligned} \right\} \Rightarrow F_{B,\min} = -\mu_x G \quad (\text{E.3})$$

Der Wert  $F_{M,\max}$  in (E.2a) lässt sich nur dann angeben, wenn das Fzg frontgetrieben ( $k_{MV} = 1$ ) oder heckgetrieben ( $k_{MV} = 0$ ) ist, weil dann jeweils ein Quotient gegen Unendlich geht. Bei Allrad-Fahrzeugen lässt sich a priori nicht sagen, welcher Wert an welcher Stelle der Strecke kleiner sein wird, weil die Kraftverteilung  $k_{MV}$  im Gegensatz zu  $k_{BV}$  nicht ideal, sondern konstant ist (zumindest in dem vorliegenden Modell). Setzt man die Normalkraft (3.51) in

$$F_M \leq f(F_{zrv}(F_M), F_{zrh}(F_M)) \Big|_{F_B=0} \quad (\text{E.4})$$

ein, wobei  $f$  die rechte Seite von (E.2a) ist, und löst (E.4) dann nach  $F_M$  auf, so ergibt sich:

$$F_M \leq \begin{cases} \mu_x \frac{m_v g c_\theta c_\phi}{k_{MV} + \mu_x \frac{h_{SP}}{l}}, & \frac{F_{z_{rv}}}{k_{MV}} \leq \frac{F_{z_{rh}}}{1 - k_{MV}} \\ \mu_x \frac{m_h g c_\theta c_\phi}{1 - k_{MV} - \mu_x \frac{h_{SP}}{l}}, & \frac{F_{z_{rv}}}{k_{MV}} > \frac{F_{z_{rh}}}{1 - k_{MV}} \end{cases} \quad (E.5)$$

Doch die Fallunterscheidung in (E.5) ist weiterhin unklar, was noch deutlicher wird, wenn man sie so umformt, dass die Unbekannte  $F_M$  auf einer Seite der Ungleichung separiert ist:

$$\begin{aligned} \frac{F_{z_{rv}}}{k_{MV}} &\leq \frac{F_{z_{rh}}}{1 - k_{MV}} \\ \frac{m_v g c_\theta c_\phi - \frac{h_{SP}}{l} F_M}{k_{MV}} &\leq \frac{m_h g c_\theta c_\phi + \frac{h_{SP}}{l} F_M}{1 - k_{MV}} \\ \left( \frac{m_v g}{k_{MV}} - \frac{m_h g}{1 - k_{MV}} \right) c_\theta c_\phi &\leq \left( \frac{1}{k_{MV}} + \frac{1}{1 - k_{MV}} \right) \frac{h_{SP}}{l} F_M \\ ((1 - k_{MV}) m_v g - k_{MV} m_h g) c_\theta c_\phi &\leq \frac{h_{SP}}{l} F_M \end{aligned} \quad (E.6)$$

Somit hängt die Fallunterscheidung von der Optimierungsvariable  $F_M$  selbst ab. Lediglich unter der Annahme  $h_{SP} = 0$  ergibt sich die einfache Fallunterscheidung

$$(1 - k_{MV}) m_v g \leq k_{MV} m_h g. \quad (E.7)$$

Es bleibt daher i. A. nur die Option, beide Fälle von (E.5) zu berechnen und das Maximum davon als  $F_{M,max}$  anzusetzen, was in Anbetracht dessen, dass diese Box-Restriktion ohnehin nur als zusätzliche Grenze gedacht ist, keinen wesentlichen Nachteil darstellt.

# Literaturverzeichnis

- [1] Adamy, J. *Nichtlineare Systeme und Regelungen*. Berlin, Heidelberg: Springer, 2018. DOI: 10.1007/978-3-662-55685-6 (Referenziert auf S. 6, 10).
- [2] Allgöwer, F. und Gilles, E. D. „Nichtlinearer Reglerentwurf auf der Grundlage exakter Linearisierungstechniken“. In: *Nichtlineare Regelung*. Bd. 1026. VDI-Berichte. Düsseldorf: VDI-Verl., 1993, S. 209–234. URL: <https://d-nb.info/1023682397/34> (Referenziert auf S. 5, 6).
- [3] Andersson, J. A. E. et al. „CasADi. A software framework for nonlinear optimization and optimal control“. In: *Mathematical Programming Computation* 11.1 (2019), S. 1–36. DOI: 10.1007/s12532-018-0139-4 (Referenziert auf S. 85).
- [4] Bainbridge, L. „Ironies of Automation“. In: *Automatica* 19.6 (1983), S. 775–779. DOI: 10.1016/0005-1098(83)90046-8 (Referenziert auf S. 1, 2).
- [5] Bauer, M. *Methoden zur modellbasierten Fahrdynamikanalyse und Bewertung von Fahrdynamikregelsystemen*. Zugl.: Diss. TU Darmstadt, 2015. Bd. 792. Fortschrittberichte VDI: Reihe 12. Düsseldorf: VDI-Verl., 2015 (Referenziert auf S. 31, 33).
- [6] Bechtloff, J. P. *Schätzung des Schwimmwinkels und fahrdynamischer Parameter zur Verbesserung modellbasierter Fahrdynamikregelungen*. Zugl.: Diss. TU Darmstadt, 2018. Bd. 809. Fortschrittberichte VDI: Reihe 12. Düsseldorf: VDI-Verl., 2018. URL: <http://nbn-resolving.org/urn:nbn:de:tuda-tuprints-80692> (Referenziert auf S. 29, 33).
- [7] Bellman, R. *Adaptive control processes. A guided tour*. Princeton, NJ: Princeton University Press, 1961 (Referenziert auf S. 12).
- [8] Bender-Napp, A. *Zukunft des autonomen Fahrens*. Hrsg. von auto motor und sport. 3.09.2019. URL: <https://www.auto-motor-und-sport.de/news/zukunft-des-autonomen-fahrens-vw-konzernforschung-vorentwicklung/> (besucht am 01.11.2019) (Referenziert auf S. 97).
- [9] Bender, E. et al. „Antikollisionssystem Proreta – Der Weg zum unfallvermeidenden Fahrzeug. Teil 2: Ergebnisse“. In: *ATZ - Automobiltechnische Zeitschrift* 109.5 (2007), S. 456–463. DOI: 10.1007/BF03221892 (Referenziert auf S. 2).
- [10] Betz, J. et al. „What can we learn from autonomous level-5 motorsport?“ In: *9th International Munich Chassis Symposium 2018*. Hrsg. von Pfeffer, P. Proceedings. Wiesbaden: Springer Fachmedien, 2019, S. 123–146. DOI: 10.1007/978-3-658-22050-1\_12 (Referenziert auf S. 3).
- [11] Beucher, O. *Signale und Systeme: Theorie, Simulation, Anwendung*. 3. Aufl. Berlin, Heidelberg: Springer, 2019 (Referenziert auf S. 40).

- [12] Braghin, F. et al. „Race driver model“. In: *Computers & Structures* 86.13-14 (2008), S. 1503–1516. DOI: 10.1016/j.compstruc.2007.04.028 (Referenziert auf S. 14, 15, 17, 21, 23, 110, 114, 122).
- [13] Brayshaw, D. „The use of numerical optimisation to determine on-limit handling behaviour of race cars“. PhD Thesis. Cranfield University, 2004. URL: <https://core.ac.uk/download/pdf/139607.pdf> (Referenziert auf S. 20, 21, 38).
- [14] Brayshaw, D. und Harrison, M. F. „A quasi steady state approach to race car lap simulation in order to understand the effects of racing line and centre of gravity location“. In: *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 219.6 (2005), S. 725–739. DOI: 10.1243/095440705X11211 (Referenziert auf S. 15, 20, 21, 23, 38, 84).
- [15] Breuer, S. und Rohrbach-Kerl, A. *Fahrzeugdynamik. Mechanik des bewegten Fahrzeugs*. 1. Aufl. Wiesbaden: Springer Fachmedien, 2015. DOI: 10.1007/978-3-658-09475-1 (Referenziert auf S. 15, 39–41, 84, 123).
- [16] Bronstein, I. N. et al. *Taschenbuch der Mathematik*. 6. Aufl. Frankfurt am Main: Harri Deutsch, 2005 (Referenziert auf S. 11, 36, 152).
- [17] Burckhardt, M. *Fahrwerktechnik: Bremsdynamik und Pkw-Bremsanlagen. Bremsvorgang, Bremsstabilität, Kräfte, Fahrzeug-, Fahrbahn- und Reifeneinfluß, Bremsanlage*. 1. Aufl. Vogel-Fachbuch Kraftfahrzeugtechnik. Würzburg: Vogel, 1991 (Referenziert auf S. 42).
- [18] Burckhardt, M. und Reimpell, J., Hrsg. *Fahrwerktechnik*. 1. Aufl. Vogel-Fachbuch Kraftfahrzeugtechnik. Würzburg: Vogel, 1993 (Referenziert auf S. 31).
- [19] Burylov, I. et al. „Intel Performance Libraries: Multi-Core-Ready Software for Numeric-Intensive Computation“. In: *Intel Technology Journal* 11.04 (2007), S. 299–308. URL: [www.intel.com/content/dam/www/public/us/en/documents/research/2007-vol11-iss-4-intel-technology-journal.pdf](http://www.intel.com/content/dam/www/public/us/en/documents/research/2007-vol11-iss-4-intel-technology-journal.pdf) (Referenziert auf S. 88).
- [20] Casanova, D. „On Minimum Time Vehicle Manoeuvring. The Theoretical Optimal Lap“. PhD Thesis. Cranfield University, 2000. URL: <http://hdl.handle.net/1826/1091> (Referenziert auf S. 11, 18–25, 38, 65, 141).
- [21] Cormen, T. H. et al. *Algorithmen - Eine Einführung*. Berlin, Boston: De Gruyter, 2017. DOI: 10.1515/9783110522013 (Referenziert auf S. 100).
- [22] Donges, E. „Fahrerverhaltensmodelle“. In: *Handbuch Fahrerassistenzsysteme*. Hrsg. von Winner, H. et al. 3. Aufl. Wiesbaden: Springer Fachmedien, 2015, S. 17–26. DOI: 10.1007/978-3-658-05734-3\_2 (Referenziert auf S. 6).
- [23] Einsle, S. „Analyse und Modellierung des Reifenübertragungsverhaltens bei transienten und extremen Fahrmanövern“. Diss. TU Dresden, 2010. URL: <http://nbn-resolving.de/urn:nbn:de:bsz:14-qucosa-65158> (Referenziert auf S. 37).

- [24] Elbanhawi, M., Simic, M. und Jazar, R. „In the Passenger Seat. Investigating Ride Comfort Measures in Autonomous Cars“. In: *IEEE Intelligent Transportation Systems Magazine* 7.3 (2015), S. 4–17. DOI: 10.1109/MITS.2015.2405571 (Referenziert auf S. 95).
- [25] Elektrobit Automotive GmbH. *EB Assist ADTF*. Version 2. URL: <https://www.elektrobit.com/products/automated-driving/eb-assist/adtf/> (besucht am 08.01.2018) (Referenziert auf S. 5).
- [26] Ersoy, M. und Gies, S., Hrsg. *Fahrwerkhandbuch. Grundlagen - Fahrdynamik - Fahrverhalten - Komponenten - Elektronische Systeme - Fahrerassistenz - Autonomes Fahren - Perspektiven*. ATZ / MTZ-Fachbuch. Wiesbaden: Springer Fachmedien, 2017 (Referenziert auf S. 37).
- [27] Festner, M., Eicher, A. und Schramm, D. „Beeinflussung der Komfort- und Sicherheitswahrnehmung beim hochautomatisierten Fahren durch fahrfremde Tätigkeiten und Spurwechseldynamik“. In: *11. Workshop Fahrerassistenzsysteme und automatisiertes Fahren*. Hrsg. von Bengler, K. Darmstadt: Uni-DAS e.V., 2017, S. 63–73. URL: <https://www.uni-das.de/images/pdf/veroeffentlichungen/2017/07.pdf> (besucht am 03.12.2019) (Referenziert auf S. 95).
- [28] Fletcher, R. und Leyffer, S. „Nonlinear programming without a penalty function“. In: *Mathematical Programming* 91.2 (2002), S. 239–269. DOI: 10.1007/s101070100244 (Referenziert auf S. 61, 62).
- [29] Föllinger, O. et al. *Regelungstechnik. Einführung in die Methoden und ihre Anwendung*. 12., überarbeitete Auflage. Lehrbuch Studium. Berlin und Offenbach: VDE Verlag GmbH, 2016 (Referenziert auf S. 6, 7, 125).
- [30] *Formula Student Germany 2018. Internationaler Konstruktionswettbewerb*. ATZ extra Sonderheft. Bd. 8.2018. Wiesbaden: Springer Vieweg (Referenziert auf S. 3).
- [31] Gadola, M. et al. „A Tool for Lap Time Simulation“. In: *Motorsports Engineering Conference & Exposition*. SAE Technical Paper Series. Warrendale, PA, United States: SAE International 400 Commonwealth Drive, 1996. DOI: 10.4271/962529 (Referenziert auf S. 11, 19, 23).
- [32] Gerdts, M. et al. „Generating locally optimal trajectories for an automatically driven car“. In: *Optimization and Engineering* 10.4 (2009), S. 439–463. DOI: 10.1007/s11081-008-9047-1 (Referenziert auf S. 21, 23, 25, 38).
- [33] Graichen, K. *Methoden der Optimierung und optimalen Steuerung. Vorlesungsskript Wintersemester 2017/18*. Hrsg. von Graichen, K. Ulm, 2017. URL: [https://www.uni-ulm.de/fileadmin/website\\_uni\\_ulm/iui.inst.110/Downloads/Vorlesung/Optimierung/Skript/Skript\\_MOOS\\_WS1718.pdf](https://www.uni-ulm.de/fileadmin/website_uni_ulm/iui.inst.110/Downloads/Vorlesung/Optimierung/Skript/Skript_MOOS_WS1718.pdf) (besucht am 04.02.2019) (Referenziert auf S. 49).
- [34] Gutjahr, B. und Werling, M. „Optimale Fahrzeugquerführung mittels linearer, zeitvarianter MPC“. In: *10. Workshop Fahrerassistenzsysteme*. Hrsg. von Stiller, C. Darmstadt: Uni-DAS e.V., 2015, S. 61–70 (Referenziert auf S. 18, 22, 27, 132, 138).

- [35] Hansen, T. *Kombinierte Längs- und Quertrajektorienplanung für automatisierte Fahrstreifenwechsel*. Zugl.: Diss., TU Darmstadt, 2018. 1. Aufl. Berlin: epubli, 2018. URL: <https://tuprints.ulb.tu-darmstadt.de/8081/> (Referenziert auf S. 11, 12).
- [36] Heilmeier, A., Graf, M. und Lienkamp, M. „A Race Simulation for Strategy Decisions in Circuit Motorsports“. In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, S. 2986–2993. DOI: 10.1109/ITSC.2018.8570012 (Referenziert auf S. 14).
- [37] Heilmeier, A. et al. „Minimum curvature trajectory planning and control for an autonomous race car“. In: *Vehicle System Dynamics* 25.8 (2019), S. 1–31. DOI: 10.1080/00423114.2019.1631455 (Referenziert auf S. 23, 24, 110, 112, 114–118, 121, 122).
- [38] Hendriks, J. P. M., Meijlink, T. J. J. und Kriens, R. F. C. „Application of Optimal Control Theory to Inverse Simulation of Car Handling“. In: *Vehicle System Dynamics* 26.6 (1996), S. 449–461. DOI: 10.1080/00423119608969319 (Referenziert auf S. 19, 23).
- [39] Higham, N. J. *Accuracy and stability of numerical algorithms*. 2. Aufl. Philadelphia, Pa.: Society for Industrial and Applied Mathematics, 2002. DOI: 10.1137/1.9780898718027 (Referenziert auf S. 147).
- [40] Hochrein, P. *Leistungsoptimale Regelung von Hochstromverbrauchern im Fahrwerk*. Zugl.: Diss. Universität Kassel, 2013. Kassel University Press, 2013. URL: <http://www.uni-kassel.de/upress/online/frei/978-3-86219-604-3.volltext.frei.pdf> (Referenziert auf S. 34).
- [41] Hohage, T. und Rapin, G. *Numerische Behandlung partieller Differentialgleichungen. Skriptum SS 2005, Universität Göttingen*. 2005 (Referenziert auf S. 71).
- [42] Hyman, J. M. „Accurate Monotonicity Preserving Cubic Interpolation“. In: *SIAM Journal on Scientific and Statistical Computing* 4.4 (1983), S. 645–654. DOI: 10.1137/0904045 (Referenziert auf S. 103, 108).
- [43] IEEE Computer Society. *IEEE Standard for Floating-Point Arithmetic*. IEEE 754. New York, 2008. DOI: 10.1109/IEEESTD.2008.4610935 (Referenziert auf S. 147).
- [44] Intel Corporation. *Intel Math Kernel Library (MKL)*. Version 2019.3.203. 2019. URL: <https://software.intel.com/en-us/mkl> (besucht am 03.06.2019) (Referenziert auf S. XII, 88, 103, 126).
- [45] Isermann, R., Hrsg. *Fahrdynamik-Regelung. Modellbildung, Fahrerassistenzsysteme, Mechatronik*. 1. Aufl. Wiesbaden: Vieweg, 2006 (Referenziert auf S. 31).
- [46] Isidori, A. *Nonlinear control systems*. 3. Aufl. Communications and control engineering series. Berlin: Springer, 1995 (Referenziert auf S. 6).
- [47] Ito, T., Takata, A. und Oosawa, K. „Time Required for Take-over from Automated to Manual Driving“. In: *SAE 2016 World Congress and Exhibition*. SAE Technical Paper Series. SAE International 400 Commonwealth Drive, Warrendale, PA, United States, 2016. DOI: 10.4271/2016-01-0158 (Referenziert auf S. 1).



- [48] Kapania, N. R. „Trajectory planning and control for an autonomous race vehicle“. PhD Thesis. Stanford University, 2016 (Referenziert auf S. 15, 16, 22, 23).
- [49] Kelly, D. P. „Lap time simulation with transient vehicle and tyre dynamics“. PhD Thesis. Cranfield University, 2008. URL: <http://dspace.lib.cranfield.ac.uk/handle/1826/4791> (Referenziert auf S. 21, 22, 25).
- [50] Lima, P. F. et al. „Progress Maximization Model Predictive Controller“. In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, S. 1075–1082. DOI: 10.1109/ITSC.2018.8569647 (Referenziert auf S. 18, 22–24, 84).
- [51] Löfberg, J. „YALMIP. A toolbox for modeling and optimization in MATLAB“. In: *IEEE International Symposium on Computer Aided Control Systems Design*. 2004, S. 284–289. DOI: 10.1109/CACSD.2004.1393890. URL: <https://yalmip.github.io/> (besucht am 03.06.2019) (Referenziert auf S. 85).
- [52] Mehrotra, S. „On the Implementation of a Primal-Dual Interior Point Method“. In: *SIAM Journal on Optimization* 2.4 (1992), S. 575–601. DOI: 10.1137/0802028 (Referenziert auf S. 62, 63).
- [53] Merziger, G. und Wirth, T. *Repetitorium der höheren Mathematik*. 5. Aufl. Springer: Binomi-Verlag, 2006 (Referenziert auf S. 29, 111).
- [54] Mitschke, M. „Das Einspurmodell von Rieker-Schunck“. In: *ATZ - Automobiltechnische Zeitschrift* 107.11 (2005), S. 1030–1031. DOI: 10.1007/BF03223515 (Referenziert auf S. 34).
- [55] Mitschke, M. und Wallentowitz, H. *Dynamik der Kraftfahrzeuge*. 5. Aufl. Wiesbaden: Springer Fachmedien, 2014. DOI: 10.1007/978-3-658-05068-9 (Referenziert auf S. 29, 37, 40, 41).
- [56] Müller, N., Mühlmeier, M. und Martignoni, D. „Ideallinie für zeitoptimales Befahren einer Rennstrecke“. In: *ATZ - Automobiltechnische Zeitschrift* 199.101 (1999), S. 202–204 (Referenziert auf S. 19, 23).
- [57] Nocedal, J., Wächter, A. und Waltz, R. A. „Adaptive Barrier Update Strategies for Nonlinear Interior Methods“. In: *SIAM Journal on Optimization* 19.4 (2009), S. 1674–1693. DOI: 10.1137/060649513 (Referenziert auf S. 60, 62, 63).
- [58] Nocedal, J. und Wright, S. J. *Numerical Optimization*. 2. Aufl. New York: Springer, 2006. DOI: 10.1007/978-0-387-40065-5 (Referenziert auf S. 20, 51–59, 62, 86).
- [59] Pacejka, H. B. und Besselink, I. *Tire and vehicle dynamics*. 3rd ed. Amsterdam und Boston: Elsevier/BH, 2012 (Referenziert auf S. 30–32, 47).
- [60] Papageorgiou, M., Leibold, M. und Buss, M. *Optimierung. Statische, dynamische, stochastische Verfahren für die Anwendung*. 4. Aufl. Berlin, Heidelberg: Springer, 2015. DOI: 10.1007/978-3-662-46936-1 (Referenziert auf S. 10, 48, 50–53, 56–58, 62).

- [61] Perantoni, G. und Limebeer, D. J. N. „Optimal control for a Formula One car with variable parameters“. In: *Vehicle System Dynamics* 52.5 (2014), S. 653–678. DOI: 10.1080/00423114.2014.889315 (Referenziert auf S. 18, 20, 22–25, 38, 39, 117).
- [62] Pfromm, M. „Menschbezogener Umgang mit Systemstörungen bei teilautomatisierter manöverbasierter Fahrzeugführung“. Diss. TU Darmstadt, 2016. URL: <https://tuprints.ulb.tu-darmstadt.de/5724/> (Referenziert auf S. 1, 2).
- [63] Rathgeber, C. „Trajektorienplanung und -folgeregelung für assistiertes bis hochautomatisiertes Fahren“. Diss. TU Berlin, 2016. DOI: 10.14279/depositonce-5506 (Referenziert auf S. 18).
- [64] Rathgeber, C., Winkler, F. und Müller, S. „Kollisionsfreie Längs- und Quertrajektorienplanung unter Berücksichtigung fahrzeugspezifischer Potenziale“. In: *at - Automatisierungstechnik* 64.1 (2016). DOI: 10.1515/auto-2015-0052 (Referenziert auf S. 12, 13).
- [65] Reams, R. „Hadamard inverses, square roots and products of almost semidefinite matrices“. In: *Linear Algebra and its Applications* 288 (1999), S. 35–43. DOI: 10.1016/S0024-3795(98)10162-3 (Referenziert auf S. 71).
- [66] Reif, K., Hrsg. *Bosch Autoelektrik und Autoelektronik*. Wiesbaden: Vieweg+Teubner, 2011. DOI: 10.1007/978-3-8348-9902-6 (Referenziert auf S. 127).
- [67] Reif, K., Hrsg. *Fahrstabilisierungssysteme und Fahrerassistenzsysteme*. BOSCH Fachinformation Automobil. Wiesbaden: Vieweg + Teubner, 2010 (Referenziert auf S. 35).
- [68] Rohloff, M. *Richtlinien für die Anlage von Autobahnen*. RAA. Ausg. 2008. Bd. 202. FGSV R1 - Regelwerke. Köln: FGSV-Verl., 2008 (Referenziert auf S. 148).
- [69] Sari, B. und Reuss, H.-C. „Fail-Operational Safety Architecture for ADAS Systems Considering Domain ECUs“. In: *WCX World Congress Experience*. SAE Technical Paper Series. SAE International 400 Commonwealth Drive, Warrendale, PA, United States, 2018. DOI: 10.4271/2018-01-1069 (Referenziert auf S. 1).
- [70] Schacher, S., Hoedt, J. und King, R. „Fahrspezifische Geschwindigkeitsprofile für die automatische oder die kooperative Fahrt“. In: *at - Automatisierungstechnik* 66.1 (2018), S. 53–65. DOI: 10.1515/auto-2017-0093 (Referenziert auf S. 3, 15, 95).
- [71] Schacher, S. et al. „Planungsadaption und Aktivierungsschranken zur Abstimmung von vertikal kooperierenden Fahrerassistenzsystemen“. In: *at - Automatisierungstechnik* 67.7 (2019), S. 557–571. DOI: 10.1515/auto-2019-0001 (Referenziert auf S. 3).
- [72] Schäuffele, J. und Zurawka, T. *Automotive Software Engineering*. Wiesbaden: Springer Fachmedien, 2016. DOI: 10.1007/978-3-658-11815-0 (Referenziert auf S. 127).
- [73] Schlag, B. und Weller, G. „Verhaltenswissenschaftliche Aspekte von Fahrerassistenzsystemen“. In: *Handbuch Fahrerassistenzsysteme*. Hrsg. von Winner, H. et al. 3. Aufl. Wiesbaden: Springer Fachmedien, 2015, S. 71–83. DOI: 10.1007/978-3-658-05734-3\_5 (Referenziert auf S. 2).

- [74] Schmä, F. und Stoll, W. „Kinetosen“. In: *HNO* 48.5 (2000), S. 346–356. DOI: 10.1007/s001060050579 (Referenziert auf S. 95).
- [75] Schramm, D., Hiller, M. und Bardini, R. *Modellbildung und Simulation der Dynamik von Kraftfahrzeugen*. 3. Aufl. Berlin, Heidelberg: Springer, 2018 (Referenziert auf S. 15, 33).
- [76] Siedersberger, K.-H. „Von der Rennstrecke auf die Straße“. In: *at - Automatisierungstechnik* 65.7 (2017). DOI: 10.1515/auto-2017-0012 (Referenziert auf S. 2).
- [77] Siegler, B., Deakin, A. und Crolla, D. „Lap Time Simulation: Comparison of Steady State, Quasi-Static and Transient Racing Car Cornering Strategies“. In: *Motorsports Engineering Conference & Exposition*. SAE Technical Paper Series. SAE International 400 Commonwealth Drive, Warrendale, PA, United States, 2000. DOI: 10.4271/2000-01-3563 (Referenziert auf S. 19).
- [78] STFC Scientific Computing Department. *Harwell Subroutine Library HSL*. Harwell Oxford. URL: <http://www.hsl.rl.ac.uk> (besucht am 05.02.2019) (Referenziert auf S. 52).
- [79] Subosits, J. K. und Gerdes, J. C. „From the Racetrack to the Road. Real-Time Trajectory Replanning for Autonomous Driving“. In: *IEEE Transactions on Intelligent Vehicles* 4.2 (2019), S. 309–320. DOI: 10.1109/TIV.2019.2904390 (Referenziert auf S. 98).
- [80] Tanenbaum, A. S. und Austin, T. *Rechnerarchitektur. Von der digitalen Logik zum Parallelrechner*. 6., aktualisierte Aufl. Bd. 4238. Informatik. Hallbergmoos: Pearson, 2014 (Referenziert auf S. 87).
- [81] Timings, J. P. und Cole, D. J. „Minimum Maneuver Time Calculation Using Convex Optimization“. In: *Journal of Dynamic Systems, Measurement, and Control* 135.3 (2013), S. 31015-1–31015-9. DOI: 10.1115/1.4023400 (Referenziert auf S. 14, 18, 21–23, 25, 38).
- [82] Vanderbei, R. J. und Shanno, D. F. „An Interior-Point Algorithm for Nonconvex Nonlinear Programming“. In: *Computational Optimization and Applications* 13.1/3 (1999), S. 231–252. DOI: 10.1023/A:1008677427361 (Referenziert auf S. 62).
- [83] Wächter, A. „Short Tutorial. Getting Started With Ipopt in 90 Minutes“. In: *Combinatorial Scientific Computing*. Hrsg. von Naumann, U. et al. Dagstuhl Seminar Proceedings. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Deutschland, 2009. URL: <http://drops.dagstuhl.de/opus/volltexte/2009/2089> (Referenziert auf S. 61, 62).
- [84] Wächter, A. und Biegler, L. T. „Line Search Filter Methods for Nonlinear Programming. Motivation and Global Convergence“. In: *SIAM Journal on Optimization* 16.1 (2005), S. 1–31. DOI: 10.1137/S1052623403426556 (Referenziert auf S. 62).
- [85] Wächter, A. und Biegler, L. T. „Line Search Filter Methods for Nonlinear Programming. Local Convergence“. In: *SIAM Journal on Optimization* 16.1 (2005), S. 32–48. DOI: 10.1137/S1052623403426544 (Referenziert auf S. 62).

- [86] Wächter, A. und Biegler, L. T. „On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming“. In: *Mathematical Programming* 106.1 (2006), S. 25–57. DOI: 10.1007/s10107-004-0559-y (Referenziert auf S. XII, 13, 59–62).
- [87] Wächter, A. und Laird, C. *Ipopt*. Version 3.12.10. 2018. URL: <https://github.com/coin-or/Ipopt> (Referenziert auf S. 64).
- [88] Waldmann, P. *Entwicklung eines Fahrzeugführungssystems zum Erlernen der Ideallinie auf Rennstrecken*. Zugl.: Diss. Brandenburgische Techn. Univ. Cottbus, 2008. Bd. 1. Schriftenreihe des Lehrstuhls Fahrzeugtechnik und -antriebe der BTU Cottbus. Aachen: Shaker, 2009 (Referenziert auf S. 3, 7).
- [89] Werling, M. *Ein neues Konzept für die Trajektoriengenerierung und -stabilisierung in zeitkritischen Verkehrsszenarien*. Zugl.: Diss. KIT Karlsruhe, 2010. Print on demand. Bd. 34. Schriftenreihe des Instituts für Angewandte Informatik - Automatisierungstechnik, Universität Karlsruhe (TH). KIT Scientific Publishing, 2011. DOI: 10.5445/KSP/1000021738 (Referenziert auf S. 12, 13, 131).
- [90] Werling, M., Reinisch, P. und Gresser, K. „Kombinierte Brems-Ausweich-Assistenz mittels nichtlinearer modellprädiktiver Trajektorienplanung für den aktiven Fußgängerschutz“. In: 8. *Workshop Fahrerassistenzsysteme, FAS2012*. Hrsg. von Dietmayer, K. Darmstadt: Uni-DAS e.V., 2012, S. 77–86 (Referenziert auf S. 2, 3, 18, 66, 132, 138).
- [91] Winner, H. und Schopper, M. „Adaptive Cruise Control“. In: *Handbuch Fahrerassistenzsysteme*. Hrsg. von Winner, H. et al. 3. Aufl. Wiesbaden: Springer Fachmedien, 2015, S. 851–891. DOI: 10.1007/978-3-658-05734-3\_46 (Referenziert auf S. 1).
- [92] Winner, H. et al. „Grundlagen von Frontkollisionsschutzsystemen“. In: *Handbuch Fahrerassistenzsysteme*. Hrsg. von Winner, H. et al. 3. Aufl. Wiesbaden: Springer Fachmedien, 2015, S. 893–912. DOI: 10.1007/978-3-658-05734-3\_47 (Referenziert auf S. 133).

## Eigene Veröffentlichungen

- [93] Gundlach, I. *Modellbasierte Trajektorienplanung für automatisiertes Fahren im Grenzbereich*. 52. Regelungstechnisches Kolloquium 2018. Boppard, 21. – 23.02.2018.
- [94] Gundlach, I. *Rennlinienberechnung für automatisiertes Fahren im fahrdynamischen Grenzbereich*. Regelungstechnisches Seminar am Lehrstuhl für Regelungstechnik der TU München. 11.07.2018.
- [95] Gundlach, I. und Konigorski, U. „Modellbasierte Online-Trajektorienplanung für zeitoptimale Rennlinien“. In: *at - Automatisierungstechnik* 67.9 (2019), S. 799–813. DOI: 10.1515/auto-2019-0032 (Referenziert auf S. 16, 26, 66, 70, 135).
- [96] Gundlach, I., Konigorski, U. und Hoedt, J. „Zeitoptimale Trajektorienplanung für automatisiertes Fahren im fahrdynamischen Grenzbereich. Eine modellbasierte Rundenzeitoptimierung für seriennahe Fahrzeuge“. In: *AUTOREG 2017*. Bd. 2292. VDI-Berichte. Düsseldorf: VDI-Verl., 2017, S. 223–234 (Referenziert auf S. 23, 66, 110, 116, 117).

## Betreute studentische Arbeiten

- [97] Hamacher, N. „Entwicklung eines Interfaces zur Rennlinienberechnung“. Bachelorarbeit. TU Darmstadt, 2016.
- [98] Kirchhoff, M., Reising, D. und Emmerich, S. „Trajektorienplanung für Rennstrecken mittels Optimierung“. Projektseminar. TU Darmstadt, 2018.
- [99] Lang-Lajendäcker, M. „Auswirkung der vertikalen Streckentopologie auf die maximal fahrbare Geschwindigkeit“. Bachelorarbeit. TU Darmstadt, 2018.
- [100] Lang-Lajendäcker, M. „Modellierung der Horizontalodynamik eines Fahrzeugs“. Proseminar. TU Darmstadt, 2017.
- [101] Ngueko Kameni, H. B. „Modellbasierte Planung von Ausweichtrajektorien um statische Hindernisse im fahrdynamischen Grenzbereich“. Masterarbeit. TU Darmstadt, 2017.
- [102] Rogulenko, E. R. „Berechnung zeitoptimaler Ausweichtrajektorien mittels konvexer Optimierung“. Bachelorarbeit. TU Darmstadt, 2017 (Referenziert auf S. 14).
- [103] Schiller, J. „Zeitoptimale Trajektorienplanung unter Berücksichtigung dynamischer Objekte“. Masterarbeit. TU Darmstadt, 2018 (Referenziert auf S. 146).
- [104] Spinnler, D. „Quasistatische Trajektorienplanung auf Basis krümmungsminimaler Rennlinien“. Masterarbeit. TU Darmstadt, 2017 (Referenziert auf S. 15).
- [105] Taraf, A. „Fahrzeugführung und Regelung unter besonderer Berücksichtigung von Störgrößen“. Masterarbeit. TU Darmstadt und Continental AG, Eschborn, 2016.
- [106] Tsaava, O. „Vergleich verschiedener Regelungskonzepte zur Kolonnenfahrt“. Bachelorarbeit. TU Darmstadt, 2015.